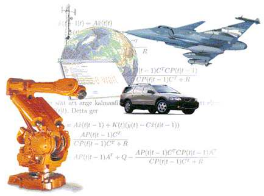
Identification of Linear and Nonlinear Dynamical Systems

Theme 3: Nonlinear Models



Lennart Ljung

Division of Automatic Control
Linköping University
Sweden

Nonlinear models - Outline

- General aspects
- Black-box models
- Grey-box models
- Special issues for non-linear models

General Aspects

Let Z^t denote all available (input-output) data up to time t. A mathematical model for the system is a function from these data to the space where the output at time t, $y(t)$ lives, in general

$$\hat{y}(t|t-1) = g(Z^{t-1}, t)$$

The function can be thought of as a predictor of the next output.
A parametric model structure is a parameterized family of such models:

$$g(Z^{t-1}, \theta)$$

All aspects on curve fitting applies pretty much also to this case. The difficulty is the enormous richness in possibilities of parameterizations. There are two main cases

- Black-box models: General models of great flexibility
- Grey-box models: Models that incorporate some knowledge of the character of the actual system.

Nonlinear models - Outline

- General aspects
- Black-box models
  - Choice of regressors and nonlinear function
  - Functions for a scalar regressor
  - Expansion into multiple regressors
  - Examples of "named" structures
- Grey-box Models
- Special issues for non-linear models

Black-box Models: General Comments

The general mapping $g(Z^{t-1}, \theta)$ is normally too flexible. Let us split it into one mapping from Z^{t-1} to a regression vector $\varphi(t)$ of fixed dimension d and a mapping g from R^d to R (assuming the output to be scalar):

$$g(Z^{t-1}, \theta) = g(\varphi(t), \theta)$$
$$\varphi(t) = \varphi(Z^{t-1}) \quad (\text{ or } \varphi(t,\theta) = \varphi(Z^{t-1}, \theta))$$

Leaves two problems

1. Choose the mapping $g(\varphi, \theta)$
2. Choose the regression vector $\varphi(t)$

NL Black Box: Choice of g

First, consider φ to be scalar. Basic form

$$g(\varphi, \theta) = \sum_{k=1}^{N} \alpha_k \kappa(\beta_k(\varphi - \gamma_k))$$

- $\kappa(x) = \cos(x)$: Fourier transform
- $\kappa(x) = U(x)$: Unit pulse, gives piecewise constant functions g.
  - Soft version: $\kappa(x) = e^{-x^2/2}$
- $\kappa(x) = H(x)$: Step at $x = 0$, gives also piecewise constant functions
  - Soft version: $\kappa(x) = \frac{1}{1+e^{-x}}$
- α coordinates, β scale or dilation, γ location

Several Regressors

Consider now φ to be a d-dimensional vector, but let still $\kappa(x)$ be a function of one variable. How to interpret $\kappa(\beta(\varphi - \gamma))$?

Radial $\beta(\varphi - \gamma) = \|\varphi - \gamma\|_\beta = (\varphi - \gamma)^T \beta (\varphi - \gamma)$
γ a d-dimensional vector, β a $d|d$-matrix (positive definite) or scaled version of the identity matrix with β a scalar.
Describes an ellipsoid in R^d.

Ridge $\beta(\varphi - \gamma) = \beta^T\varphi - \gamma$
β a d-dimensional vector, γ a scalar.
Describes a hyperplane in R^d

Tensor κ is a product of factors corresponding to the components of the vector: $\kappa(\beta(\varphi - \gamma)) = \prod_{k=1}^{d} \kappa(\beta_k(\varphi_k - \gamma_k))$
γ and β are d-dimensional vectors and subscript denotes component.

Examples of Named Structures

- ANN: artificial Neural Networks
  - One hidden layer sigmoidal: $\kappa(x) = \frac{1}{1+e^{-x}}$, ridge extension
  - Radial Basis Networks: $\kappa(x) = e^{-x^2/2}$, radial extension
- Wavelets: κ is the "mother wavelet" and $\beta_j = 2^j$, $\gamma_k = 2^{-j}k$ (double indexing) as fixed choices
- (Neuro)-Fuzzy models: κ are the membership functions, tensor expansion

Simulation and Prediction

Suppose $\varphi(t) = [y(t-1), u(t-1)]^T$
The (one-step ahead) predicted output at time for a given model θ is then

$$\hat{y}_p(t|\theta) = g([y(t-1), u(t-1)]^T, \theta)$$

It uses the previous measurement $y(t-1)$.

A tougher test is to check how the model would behave in simulation, i.e. when only the input sequence u is used. The simulated output is obtained as above, by replacing the measured output by the simulated output from the previous step:

$$\hat{y}_s(t,\theta) = g([\hat{y}_s(t-1,\theta), u(t-1)]^T, \theta)$$

Notice a possible stability problem!

Choice of Regressors

Four players:

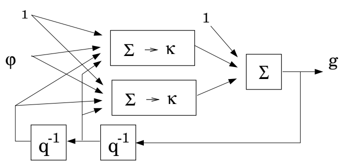
- Outputs $y(t-k)$, Inputs $u(t-k)$
- Simulated model outputs $\hat{y}_s(t-k,\theta)$
- Predicted model outputs $\hat{y}_p(t-k|\theta)$

Regressors for dynamical systems are typically chosen among those:

- NFIR-models use past inputs
- NARX-models use past inputs and outputs
- NOE-models use past inputs and past simulated outputs
- NARMAX-models use inputs, outputs predicted outputs
- NBJ-models use all four regressor types

Recurrent Networks

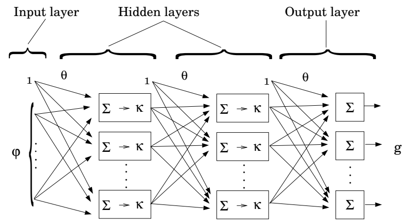
For NOE, NARMAX and NBJ, previous outputs from the model have to be fed back into the model computations on-line:



These are called recurrent networks and require considerable more computational work to fit to data.

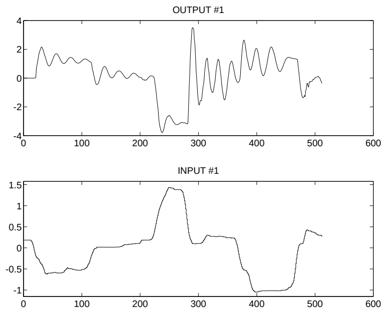
Network Aspects – Several Layers

The model structures are really basis function expansions. However, since the basis functions are variants of the same function κ, a graphical description looks like a network. One can also let the regressors be outputs from a previous layer of the network:



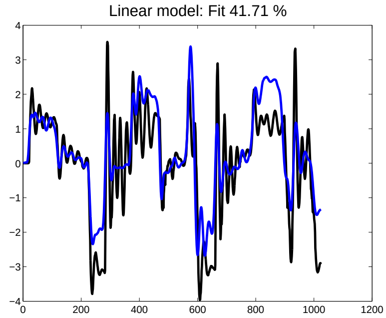
Example: Hydraulic Crane Data

These are data from a forest harvest machine:

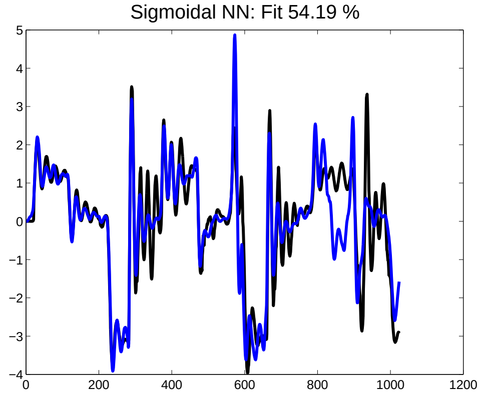


Linear Model

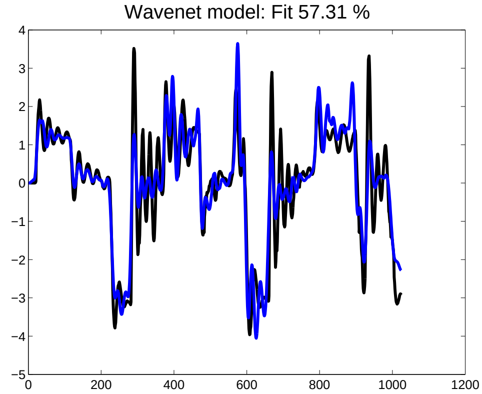
Black: Measured Output
Blue: Model Simulated Output



Sigmoidal ANN model



Wavenet Model (Radial BF ANN Model)



Nonlinear models - Outline

- General aspects
- Black-box models
- Grey-box Models
  - Physical Modeling
  - Semi-physical Modeling
  - Block-models
  - Local Linear Models
- Special issues for non-linear models

Physical Modeling

Perform physical modeling (e.g. in MODELICA) and denote unknown physical parameters by θ. Collect the model equations as

$$\dot{x}(t) = f(x(t), u(t), \theta)$$
$$y(t) = h(x(t), u(t), \theta)$$

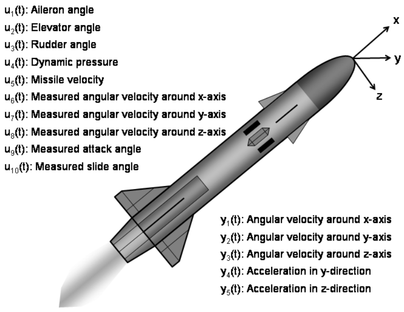
(or in DAE, Differential Algebraic Equations, form.) For each parameter θ this defines a simulated (predicted) output $\hat{y}(t|\theta)$ which is the parameterized function

$$\hat{y}(t|\theta) = g(Z^{t-1}, \theta)$$

in somewhat implicit form. To be a correct predictor this really assumes white measurement noise. Some more sophistical noise modeling is possible, usually involving *ad hoc* non-linear observers.

The approach is conceptually simple, but could be very demanding in practice.

Example: Missile



10 inputs, 5 outputs, 16 unknown parameters.

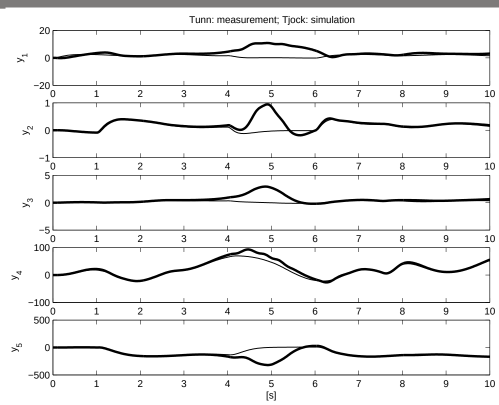
The Equations


```
function [dx, y] = missile(t, x, p, u);
MISSILE  A non-linear missile system.

 Output equation.
y = [x(1);                                         ... % Angular velocity around x
     x(2);                                         ... % Angular velocity around y
     x(3);                                         ... % Angular velocity around z
     -p(18)*u(4)*(p(1)*x(5)+p(2)*u(3))/p(22);      ... % Acceleration in y-directi
     -p(18)*u(4)*(p(3)*x(4)+p(4)*u(2))/p(22)       ... % Acceleration in z-directi
    ];

% State equations.
dx = [1/p(19)*(p(17)*p(18)*(p(5)*x(5)+0.5*p(6)*p(17)*x(1)/u(5)+   ... % Angular
           p(7)*u(1))*u(4)-(p(21)-p(20))*x(2)*x(3))+              ...
           p(23)*(u(6)-x(1));                                     ...
        1/p(20)*(p(17)*p(18)*(p(8)*x(4)+0.5*p(9)*p(17)*x(2)/u(5)+
... % Angular velocity around y-axis.
```

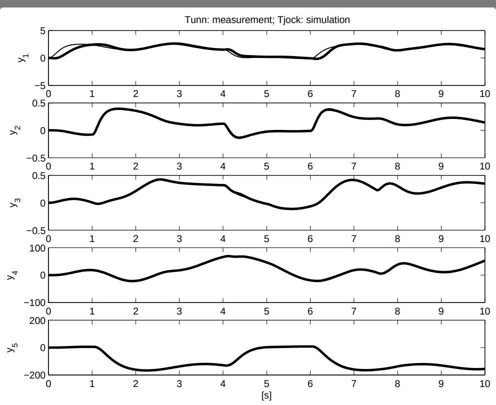

Equations ...


```
     p(10)*u(2))*u(4)-(p(19)-p(21))*x(1)*x(3))+             ...
     p(23)*(u(7)-x(2));                                     ...
  1/p(21)*(p(17)*p(18)*(p(11)*x(5)+p(12)*x(4)*x(5)+         ... % Angular
     0.5*p(13)*p(17)*x(3)/u(5)+p(14)*u(1)+                  ...
     p(15)*u(3))*u(4)-(p(20)-p(19))*x(1)*x(2))+             ...
     p(23)*(u(8)-x(3));                                     ...
  (-p(18)*u(4)*(p(3)*x(4)+p(4)*u(2)))/(p(22)*u(5))-         ... % Attack
     x(1)*x(5)+x(2)+p(23)*(u(9)/u(5)-x(4))+p(16)*x(5)^2;    ...
  (-p(18)*u(4)*(p(1)*x(5)+p(2)*u(3)))/(p(22)*u(5))-         ... % Slide a
     x(3)+x(1)*x(4)+p(23)*(u(10)/u(5)-x(5))                 ...
 ];
```


Initial Fit between Model and Data



Adjusted Fit between Model and Data

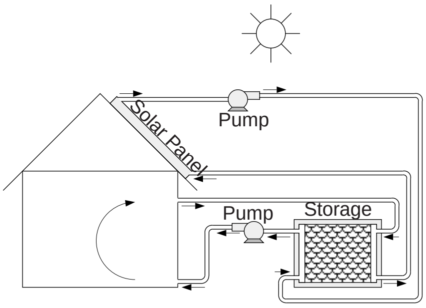


Semi-physical Models

Apply non-linear transformations to the measured data, so that the transformed data stand a better chance to describe the system in a linear relationship.
"Rules: Only high-school physics and max 10 minutes"
Simple examples: . . . .

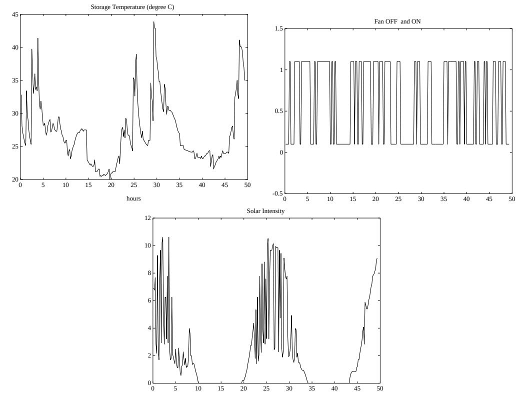
Another example:...

A Solar Heated House

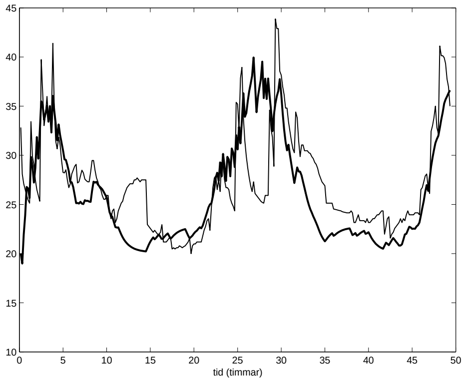


$y(t)$: temperature in storage, $I(t)$: Solar intensity $u(t)$: Pump velocity

Data



Linear Model



Think ...

Suppose we had measured the temperature $x(t)$ in the solar panel:

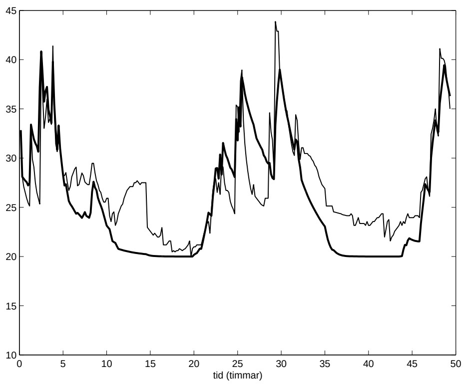
$$x(t+1) - x(t) = d_2 I(t) - d_3 x(t) - d_0 x(t) \cdot u(t)$$
$$y(t+1) - y(t) = d_0 x(t) \cdot u(t) - d_1 y(t)$$

Eliminate $x(t)$:

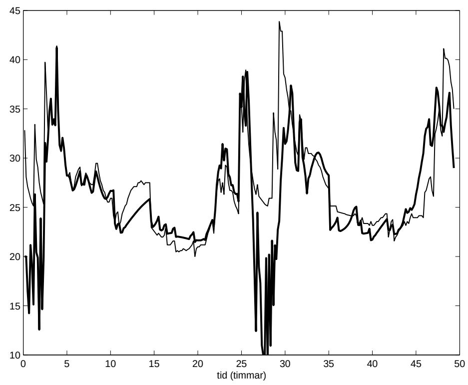
$$\begin{aligned} y(t) =& (1+d_1)y(t-1) + (1-d_3)\frac{y(t-1)u(t-1)}{u(t-2)} \\ &+ (d_3-1)(1+d_1)\frac{y(t-2)u(t-1)}{u(t-2)} + d_0 d_2 u(t-1) \cdot I(t-2) \\ &- d_0 u(t-1) y(t-1) + d_0(1+d_1)u(t-1)y(t-2) \end{aligned}$$

Reparameterize with θ being the coefficiente above, ignoring links between them.

Semi-physical Model



Neural Network Model

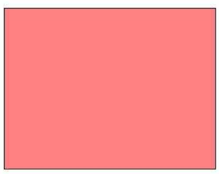


Nonlinear models - Outline

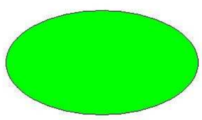
- General aspects
- Black-box models
- Grey-box Models
  - Physical Modeling
  - Semi-physical Modeling
  - Block-oriented models
  - Local Linear Models
- Special issues for non-linear models

Block-oriented Models

Building Blocks:

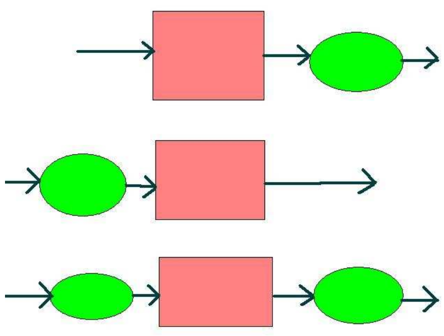


Linear Dynamic System $G(s)$



Nonlinear static function $f(u)$

Common Models

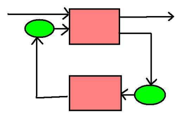


Wiener

Hammerstein

Hammerstein-Wiener

Other Combinations



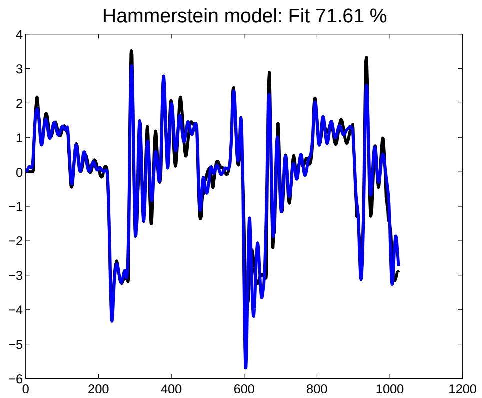
With the linear blocks parameterized as a linear dynamic system and the static blocks parameterized as a function ("curve"), this gives a parameterization of the output as

$$\hat{y}(t|\theta) = g(Z^{t-1}, \theta)$$

and the general approach of model fitting can be applied.

However, in this contexts many algorithmic variants have been suggested.

The Hydraulic Crane Again



Local Linear Models

Non-linear systems are often handled by linearization around a working point. The idea behind Local Linear Models is to deal with the nonlinearities by selecting or averaging over relevant linearized models.

Example: Tank with inflow u and outflow y and level h: Equations:

$$\dot{h} = -\sqrt{h} + u$$
$$y = \sqrt{h}$$

Linearize around level h^* with corresponding flows $u^* = y^* = \sqrt{h^*}$:

$$\dot{h} = -\frac{1}{2\sqrt{h^*}}(h - h^*) + (u - u^*)$$
$$y = y^* + \frac{1}{2\sqrt{h^*}}(h - h^*)$$

Tank Example, ctd

Sampled data model around level h^*:

$$\hat{y}_{h^*}(t) = \theta_{h^*}^T \varphi(t)$$
$$\varphi(t) = \begin{bmatrix} 1 & -y(t-T_s) & u(t-T_s) \end{bmatrix}^T \quad T_s = \text{sampling time}$$
$$\theta_{h^*} = \begin{bmatrix} \gamma^* & \alpha_{h^*} & \beta_{h^*} \end{bmatrix}^T$$

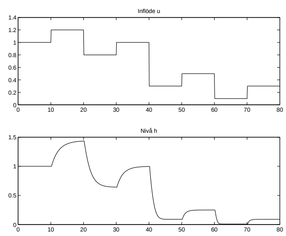
Total model: select or average over these local predictions, computed at a grid of values of h^*

$$\hat{y}(t) = \sum_{k=1}^{d} w_k(h, h_k)\hat{y}_{h_k}(t)$$

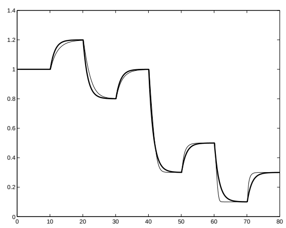
Choices of weights w_k : . . ..

Data and Linear Model

Measured data:

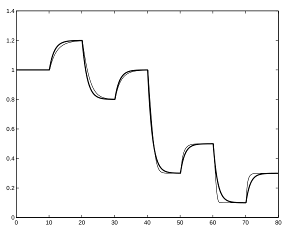


Linear Model ($d = 1$)

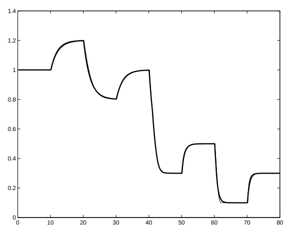


Local Linear Models

Two models (d=2)



Five models ($d = 5$)



Local Linear Models: General Comments

Let the measured working point variable (tank level in example) be denoted by $\rho(t)$ (sometimes called regime variable). If the regime variable is partitioned into d values ρ_k, the predicted output will be

$$\hat{y}(t) = \sum_{k=1}^{d} w_k(\rho(t), \rho_k)\hat{y}^{(k)}(t)$$

If the prediction $\hat{y}^{(k)}(t)$ corresponding to ρ_k is linear in the parameters, $\hat{y}^{(k)}(t) = \varphi^T(t)\theta^{(k)}$ the whole model will be a linear regression. .

To Build a Local Linear Model

To build the model, we need to

- Select the regime variable ρ
- Decide the partition of the regime variable $w_k(\rho(t),\eta)$. Here η is a parameter that describes the partition
- Find the local models in each partition.

If the local models are linear regressions, the total model will be

$$\hat{y}(t,\theta,\eta) = \sum_{k=1}^{d} w_k(\rho(t),\eta)\varphi^T(t)\theta^{(k)}$$

which for fixed η is a linear regression.

Hybrid Models and LPV Models

$$\hat{y}(t,\theta,\eta) = \sum_{k=1}^{d} w_k(\rho(t),\eta)\varphi^T(t)\theta^{(k)}$$

is also an example of a hybrid model (piecewise linear). If the partition is to be estimated too, the problem is considerably more difficult.

So called Linear Parameter Varying (LPV) are also closely related:

$$\dot{x} = A(\rho(t))x + B(\rho(t))u$$
$$y = C(\rho(t))x + D(\rho(t))u$$

Nonlinear models - Outline

- General aspects
- Black-box models
- Grey-box Models
- Special issues for non-linear models
  - Input design
  - Sparsity
  - Local minima

Experiment Design

The design of inputs for non-linear models is considerable more difficult than for linear models. For example, it is clear that a binary input will not work. (Think of a Hammerstein model!)
In addition to exciting all frequencies, an input for a general nonlinear model must also excite all amplitudes.
Gaussian noise (white or colored) would be an example of a signal that is generically exciting for nonlinear models.

Sparsity

The basic problem:

- Non-linear surfaces in high dimensions can be very complicated and need support of many observed data points.
- How to find parameterizations of such surfaces that both give a good chance of being close to the true system, and also use a moderate amount of parameters?
- The data cloud of observations is by necessity sparse in the surface space.

Sparsity: Some ideas

- Using physical insight in grey-box models is one way to allow extrapolation and interpolation in the data space on physical grounds.
- Hoping that most of the non-linear action takes place *across* hyperplanes or hyperspaces is another idea that will radically reduce the flexibility of the model.
- In statistics the problem to find such subspaces is known as the *index problem*. Finding projections S of dimension $r|m, r << m$ such that $f(\varphi) = g(S\varphi)$ captures most of nonlinearity consequently is an important problem.
- The Ridge-based neural networks can be seen as one way to find several such hyperplanes that define the structure of the non-linear effects.

Local Minima

Adjusting a parameterized model structure to data typically is a non-convex problem and several local minima of the criterion function may exist.
This is one of the most pressing problem in non-linear identification, and calls for sophisticated initialization procedures.

- In Neural Networks, some normalization is first applied to the data, and then a randomized initialization is made. Typically one will have to try several initializations
- Wavenets use an initialization based on fixed location and dilation parameters, which gives a linear regression
- For physical models, algebraic methods may produce linear regressions for initial estimates
- Block oriented models often employ several steps, fixing linear and/or nonlinear block to create smaller problems.

Conclusions Theme 3: Nonlinear Models

- A nonlinear model can be seen as nonlinear mapping from past data to the space where the output lives: $\hat{y}(t|t-1) = g(Z^{t-1}, t)$. Observations are then $y(t) = \hat{y}(t|t-1) + e(t)$.
- Useful split of mapping: $g(Z^{t-1}) = g(\varphi(Z^{t-1}))$
- Non-parametric and Parametric methods
- Black-box and Grey-box parameterizations $g(\varphi, \theta)$
- Black-box parameterizations usually employ one basic basis-function, that is scaled and located at different points
- Grey-boxes can be based on (serious) physical modeling and on more leisurely semi-physical modeling.
- Non-convexity of the optimization remains one of the more serious problems for most parametric methods.