Fault Detection Tools and Techniques

Fahmida N Chowdhury

University of Louisiana at Lafayette

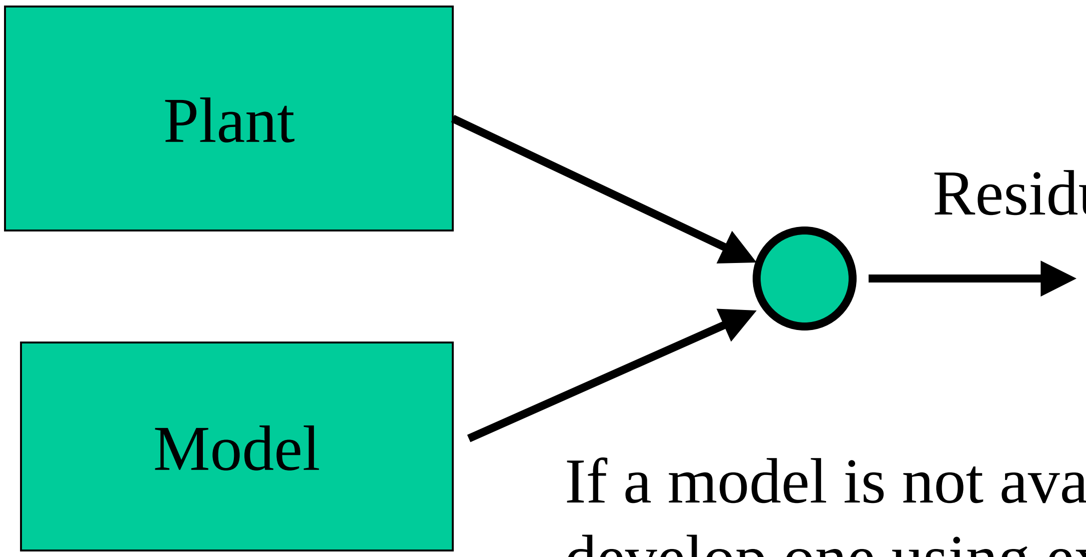
Jorge L Aravena

Louisiana State University

FAULT DETECTION

Model/Residual Based

White Box Approach



If a model is not available, must develop one using experimental data

Modeling in Real Time

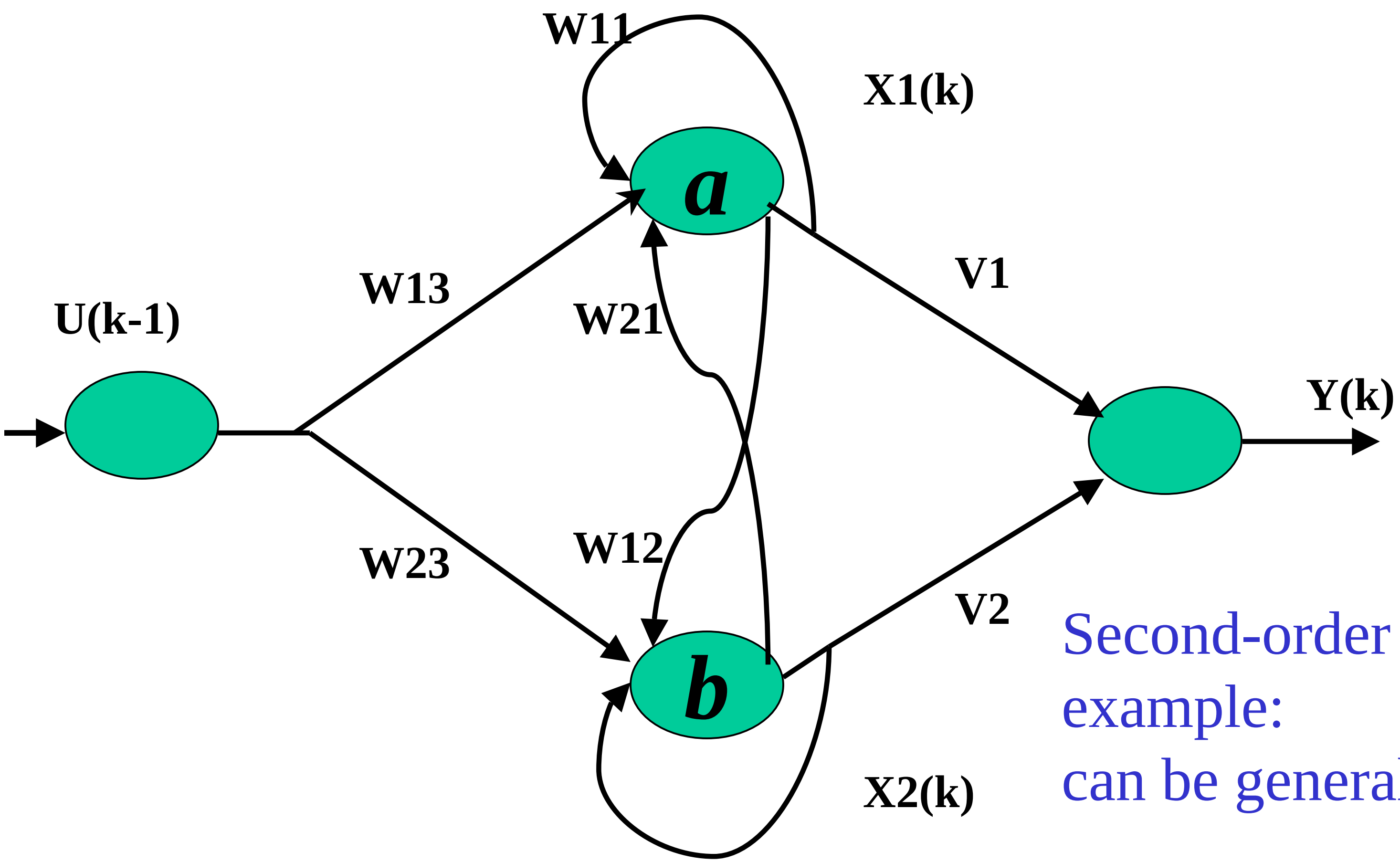
Input-output models
(ARMA, ARMAX
NARMA, NARMAX)

State-space models
[current work using time-lagged neural networks (TLRN)]

For TLRN, training issues are still open problems

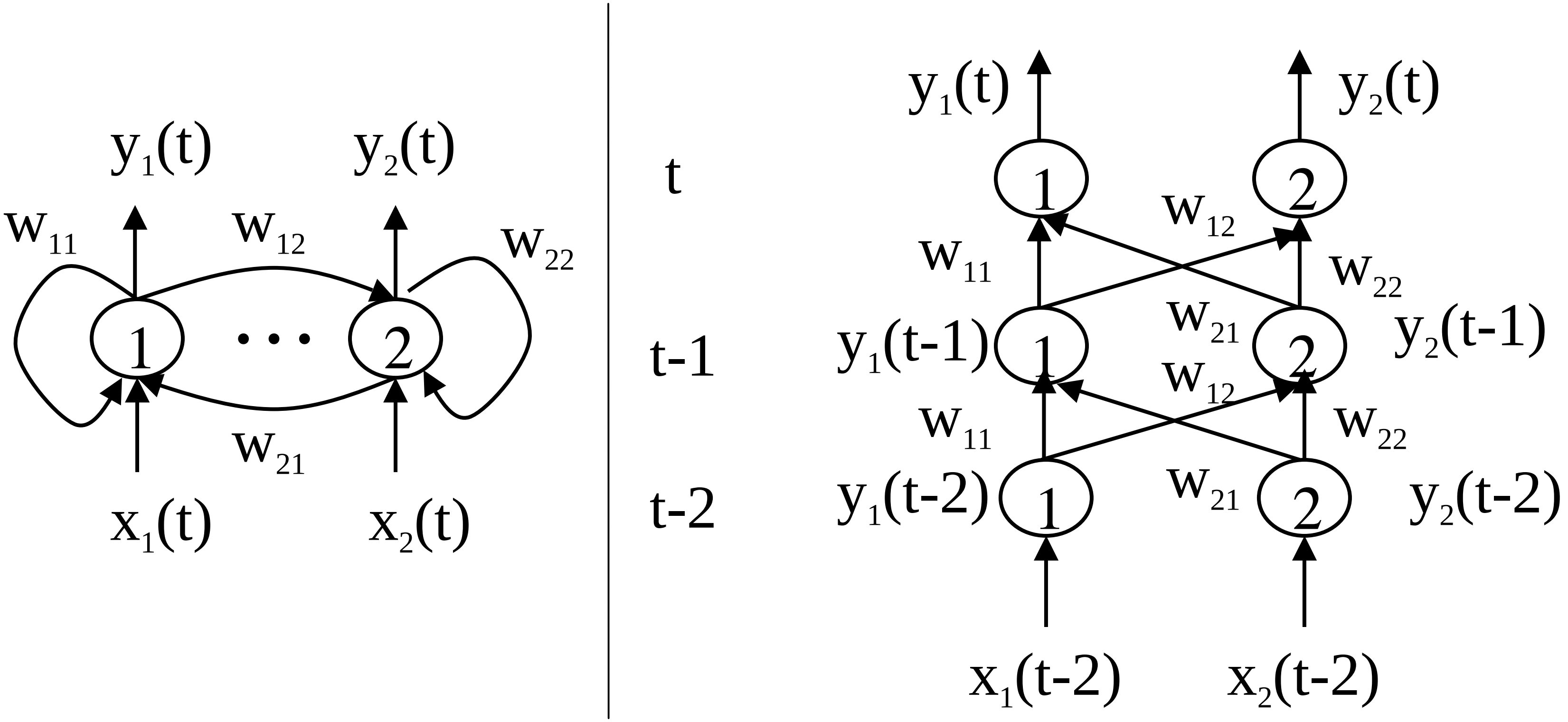
- *Backpropagation through time (BPTT)*
- *Kalman filter and EKF-based training of ANNs*

Time-lagged neural network used for state space modeling

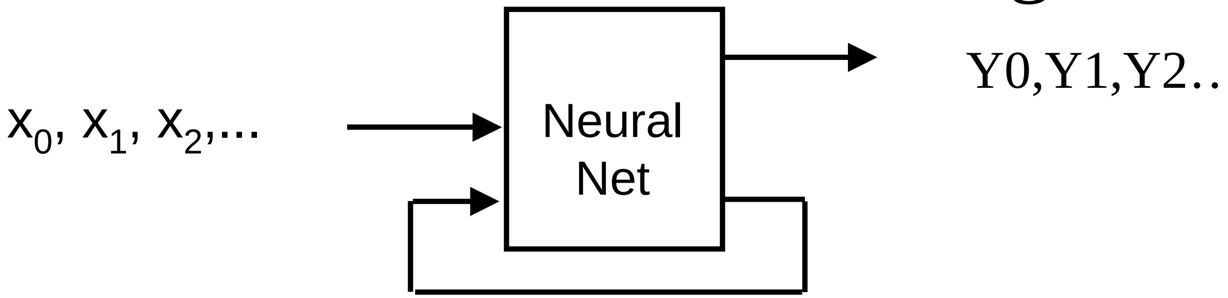


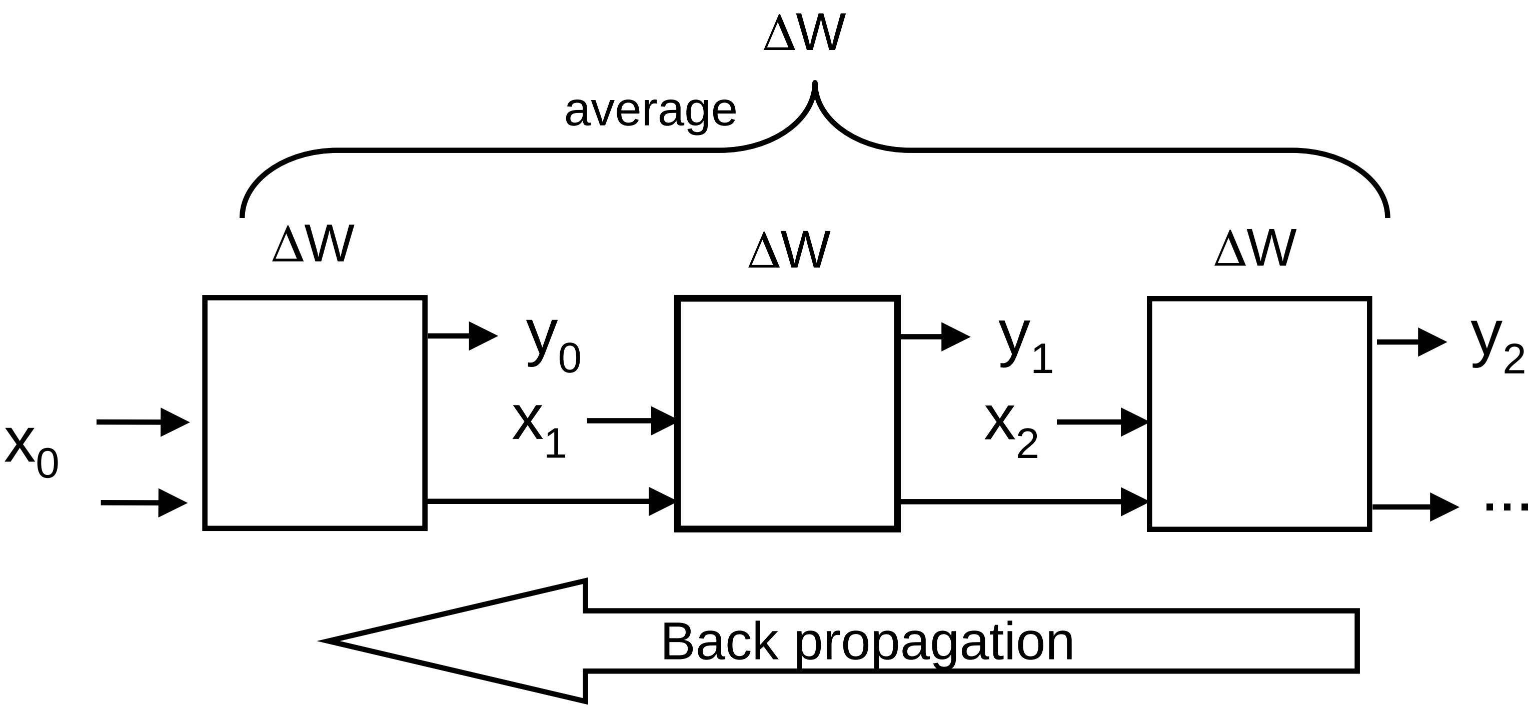
Second-order example:
can be generalized

The effect of Unfolding in BPTT



BPTT for RNN training





Extended Kalman Filter training

- EKF is a state estimation technique for nonlinear systems derived by linearizing the well-known linear-systems Kalman filter around the current estimates.

- In order to apply EKF to the task of estimating optimal weights of Recurrent Neural Networks(RNN), we interpret the weights of the network as the state of a dynamical system.

$$\mathbf{x(n+1) = f(x(n), u(n)) + q(n)} \qquad \mathbf{w(n+1) = w(n) + q(n)}$$

$$\mathbf{d(n) = h_n(x(n))} \qquad \mathbf{d(n) = h_n(w(n), u(n))}$$

w : vector containing all the weights of the RNN.

The output **d(n)** of the RNN is a function h of the weights and the input.

The NN training task now takes the form of estimating the state from an initial guess *w(0)* and the sequence of outputs and inputs: *d(0), …d(n), u(0), …, u(n).*

EKF Algorithm for RNN training

Predict | Update

(1) Project the state ahead

$$w_k^- = w_{k-1}^-$$

(2) Project the error covariance ahead

$$P_k^- = P_{k-1}$$

(1) Compute the Kalman gain

$$K = P_k^- H_k^{\,T} \left(H_k P_k^- H_k^{\,T}\right)^{-1}$$

(2) Update the estimate with measurement $\bar{z}_k$

$$\mathrm{w}_k = w_k^- + K\left(d_k - h\left(w_k^-, u(k)\right)\right)$$

(3) Update the error covariance

$$P_k = \left(I - K_k H_k\right) P_k^{\,-}$$

How good is the model that we just developed?

Testing goodness of fit:
Autocorrelation functions
Chi-squared tests
Kolmogoroff-Smirnov tests

Under no-fault conditions, in the presence of only random disturbances, the residuals must be random

Kolmogoroff-Smirnov Test

- H_0: $F(x) = F_0(x)$ True,
- H_1: false

Form the empirical estimate of $F(x)$ and use as test statistic the maximum distance between $F(x)$ and $F_0(x)$:

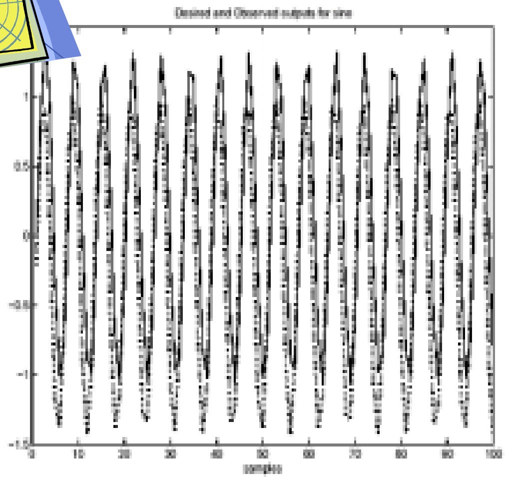
$q = \max |F(x) - F_0(x)|$

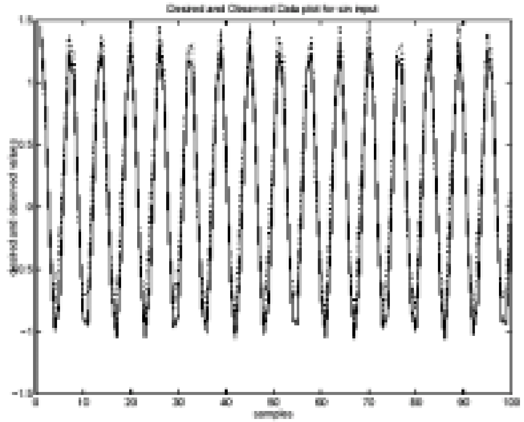
Find a constant c such that $P\{q>c|H_0\} = \alpha$

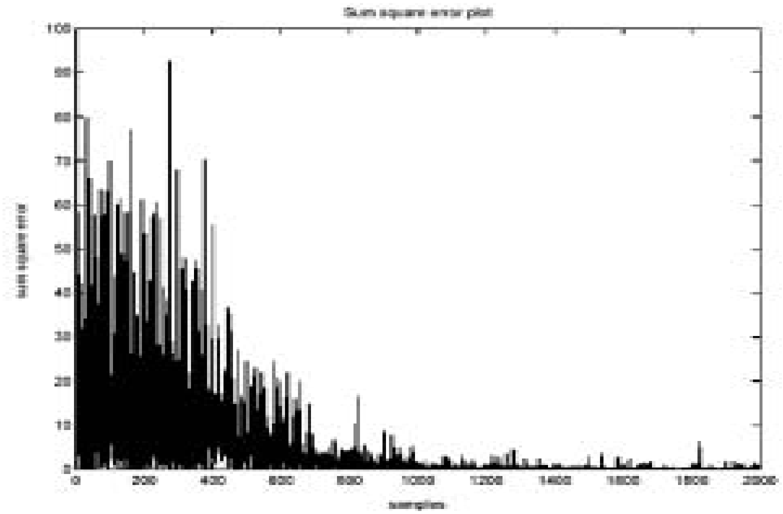
Where $\alpha = 2\exp(-2nc^2)$ (Kolmogoroff approximation)

Accept H_0 iff $q < \mathrm{sqrt}[(-1/2n) \ln(\alpha/2)] = c$

α will be the probability of false alarm (type I error)







Typical results of modeling with input-output data

For uncertain systems, it cannot be guaranteed that residuals under no-fault conditions will be random!

Many of our experimental models for arbitrary nonlinear plants showed non-perfect residuals: that is, the K-S test failed. Using these types of residuals would create many false alarms.

If there is a fault, then the residuals start to show systematic patterns

These patterns may help us classify the various faults

FAULT DETECTION

• You cannot correct what you cannot see

• At the onset of a fault normal data is nuisance

• Signal Processing can eliminate nuisance data without requiring math models

Filter Banks

Model Free/DSP Based

Black Box Approach

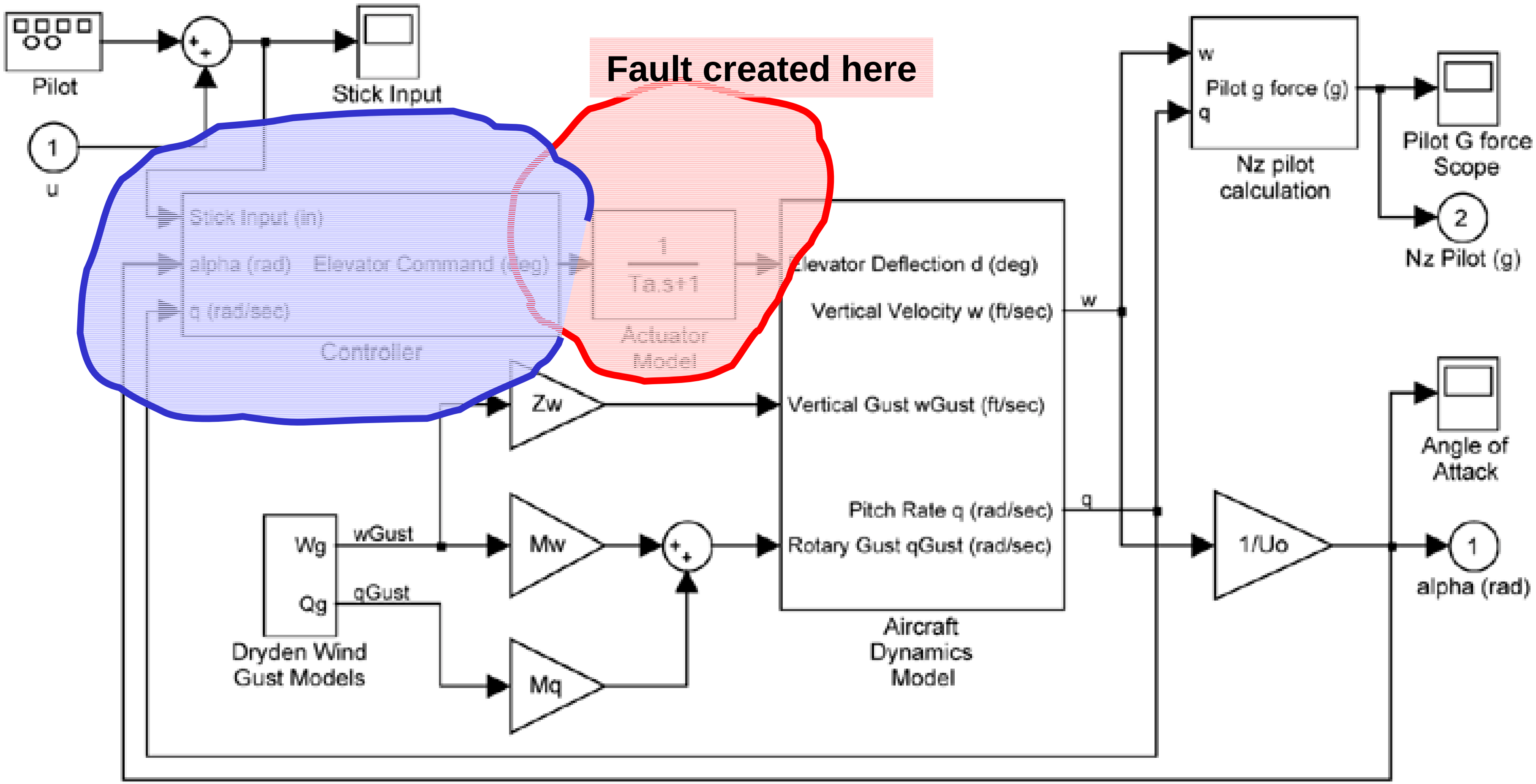
Model Free characterization in terms of changes in energy distribution

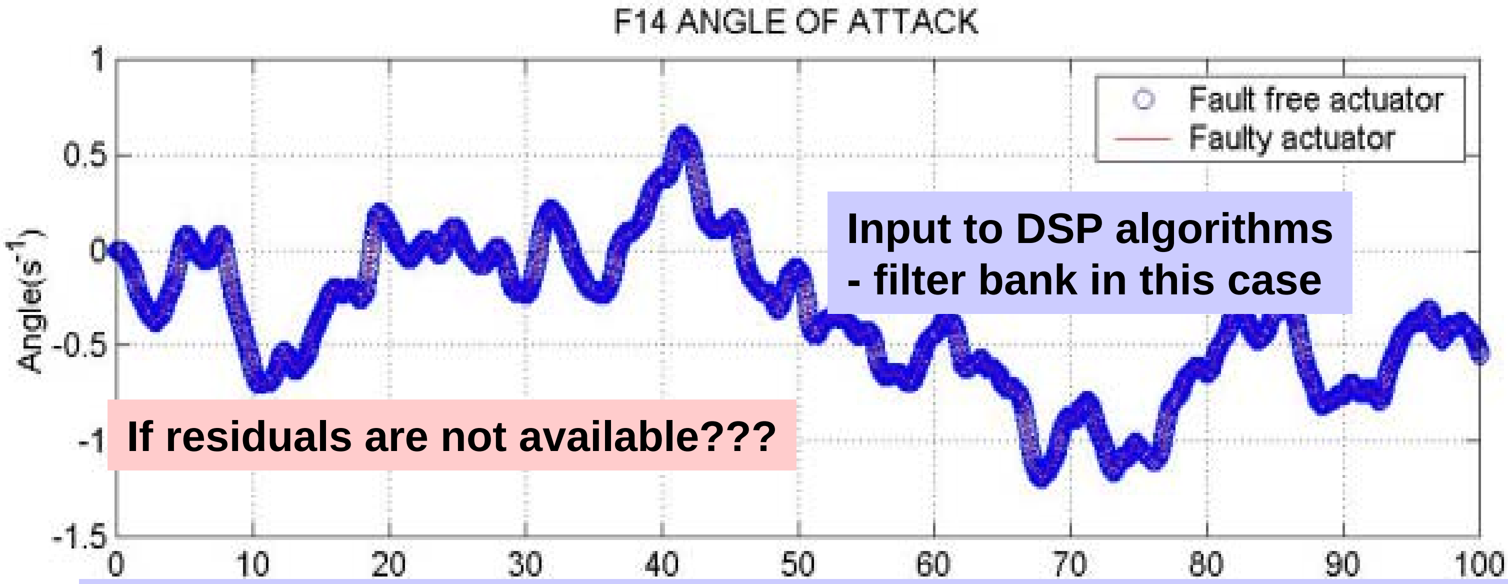
CWT & STFT

• "Enough" experimental data can replace a mathematical model

• Unsupervised clustering does not require a model

F14 SIMULINK MODEL



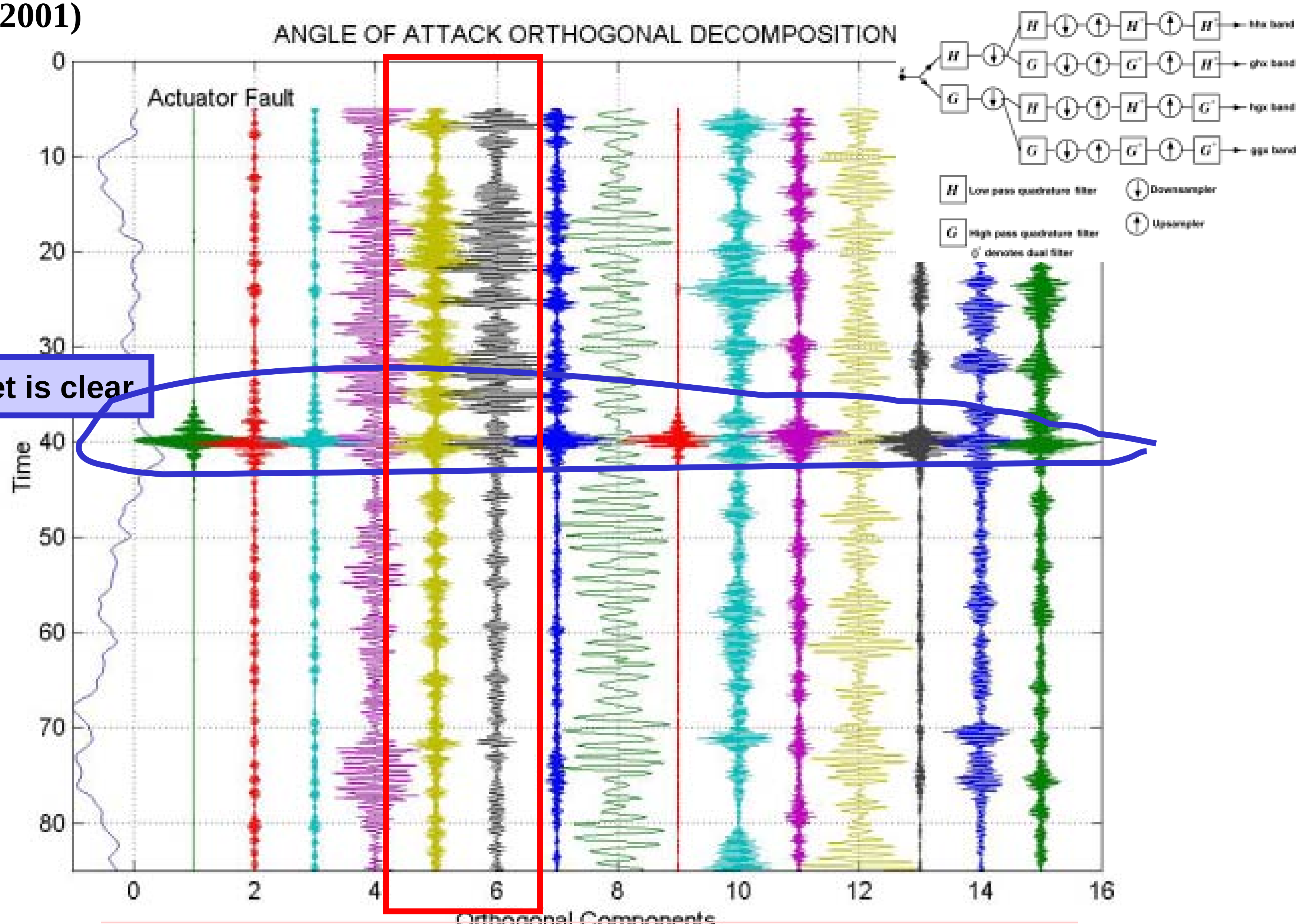


Input to DSP algorithms - filter bank in this case

If residuals are not available???

By examining the sensor reading one cannot see the onset of the fault

(DASC2001)



<u>Some</u> components show distinct pre - and post - fault behavior
It may be possible to classify faults
Faults cause changes in energy distribution

Pseudo Power Signature

- Pseudo power signature

  Develop a signature that characterizes the energy distribution of a signal in a manner that is essentially independent of the duration of the signal.

Pseudo Power Signature

- Time-frequency energy density function

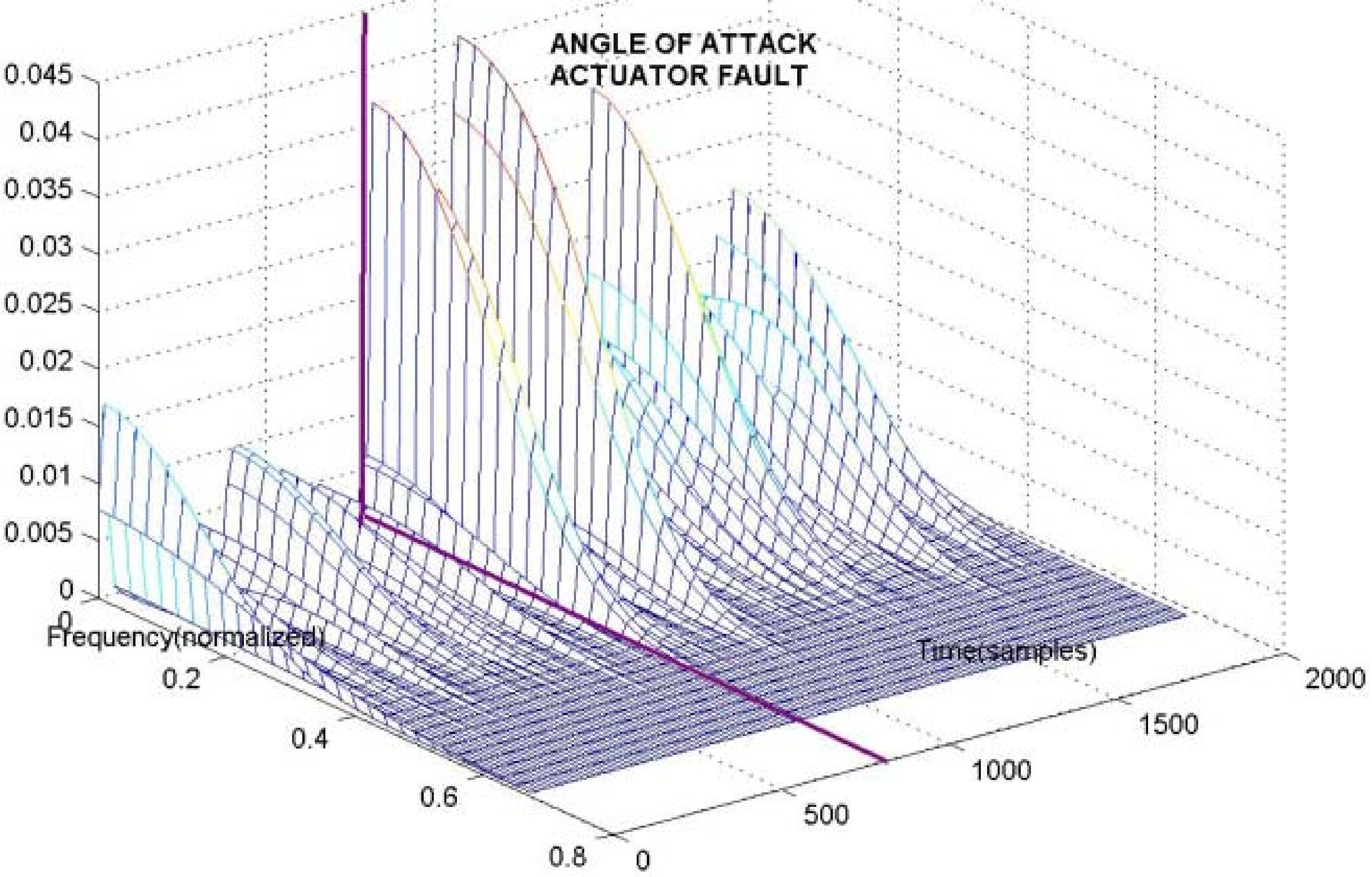
$$E_{\varphi}^{x(R)} = C_{\varphi}^{-1} \int \int_{(a,b)\in R} SC_{\varphi}^{x}(a,b) \frac{dbda}{a^2} \qquad (1)$$

$SC_{\varphi}^{x}(a,b)$ scalogram of a function with CWT $c_{\varphi}^{x}(a,b)$

$$SC_{\varphi}^{x}(a,b) = \left| c_{\varphi}^{x}(a,b) \right|^2 \qquad (2)$$

The scalogram can be used as a time-frequency energy density function.

STFT Energy distribution for details



(DASC2002)

Signature subspace

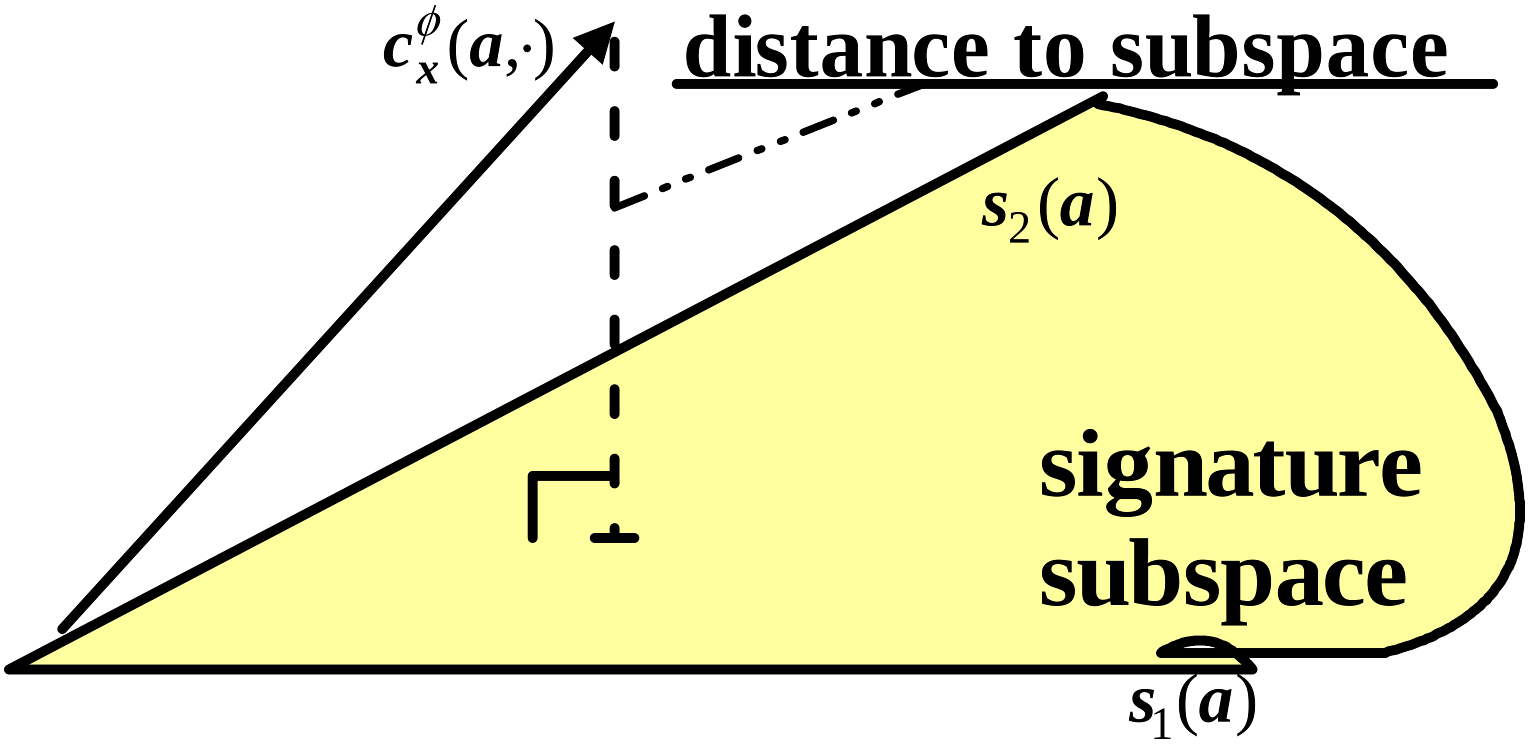
Uses Singular Value Decomposition to approximate a function of two variables

e.g. if two values are significant

$$c_x^{\phi}(a,b) = \sigma_1 s_1(a) r_1(b) + \sigma_2 s_2(a) r_2(b)$$

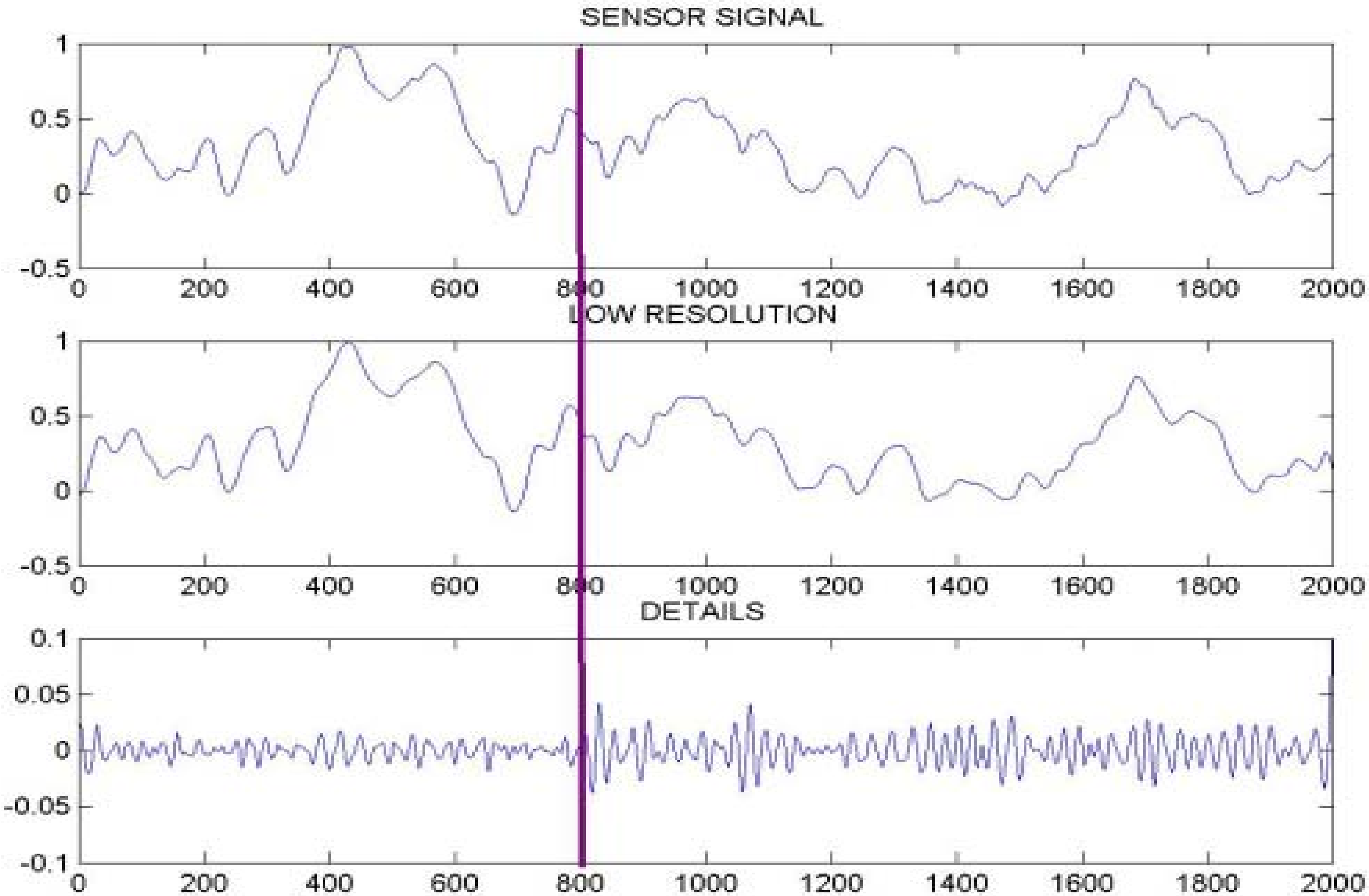
$$\Rightarrow \boldsymbol{c_x^{\phi}}(\boldsymbol{a},\cdot) \in \boldsymbol{span}\{\boldsymbol{s}_1(\boldsymbol{a}), \boldsymbol{s}_2(\boldsymbol{a})\};\ \forall \boldsymbol{b}$$

Distance Indicator



normal operation $\Rightarrow c_x^\phi(a,\cdot) \in span\{s_1(a), s_2(a)\};\ \forall b$

Data generated with 1-axis model of F14



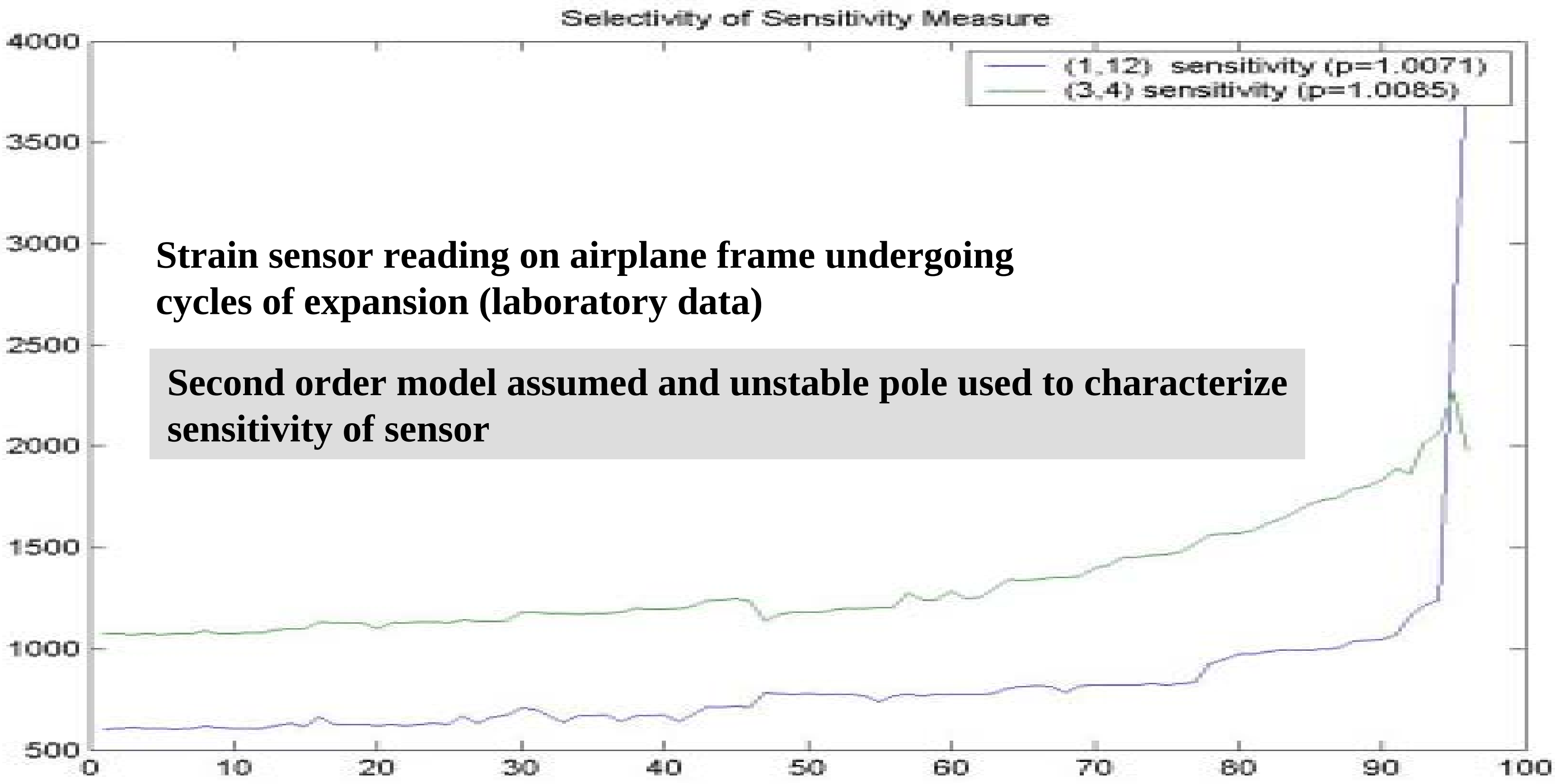
Things are beginning to get gray

If there is a fault, then the residuals start to show systematic patterns

These patterns may help us classify the various faults

Things are getting whiter

We can determine (unstable) poles of a system
Required knowledge of input: it is bounded



Strain sensor reading on airplane frame undergoing cycles of expansion (laboratory data)

Second order model assumed and unstable pole used to characterize sensitivity of sensor

And Whiter

- If the input is a stationary random process with a rational spectrum we know we can ‘whiten’ input and have an ARMAX model of the form

White sequence

$$y_k + \sum_{i=1}^{m} a_i y_{k-i} = \sum_{j=0}^{m} b_j e_{k-j}$$

$$u_k = \sum_{j=0}^{m} b_j e_{k-j}$$

$$v_n = u_{n(m+1)}$$ Uncorrelated!

$$y_{k+n(m+1)} + \sum_{i=1}^{m} a_i y_{k+n(m+1)-i} = u_n$$

Decimated output has a pure AR model

Faster than Yule-Walker

How Useful Are Residuals?

If there is no random noise in the system, residuals are very useful (FDI using parity check methods etc.)

If there is noise, residuals may be too fuzzy to use directly.

How to enhance the useful information hidden in the residuals

AutoRegressive modeling of the residuals … AR parameters estimated by a Kalman filter in real time

Using the
IFAC Benchmark Problem for FDI
Ship Propulsion System

http://www.control.auc.dk/ftc/html/body_ship_propulsion_.html

Available Models- One Engine and one propeller
Two Engines and two propellers

Detailed Description of the Benchmark available in:

•Izadi-Zamanabadi R. and M. Blanke (1999), ***A Ship Propulsion System Model for Fault-Tolerant Control***, In *Control Engineering Practice*, 7(2), 227-239.

•Izadi-Zamanabadi R. and M. Blanke (1998), ***A Ship Propulsion System as a Benchmark for Fault-tolerant Control***, Technical report, Control Engineering Dept., Aalborg University

In the ship propulsion system, we introduced a slowly developing fault in the engine torque. The output is the ship speed, and the controller output is the fuel index.

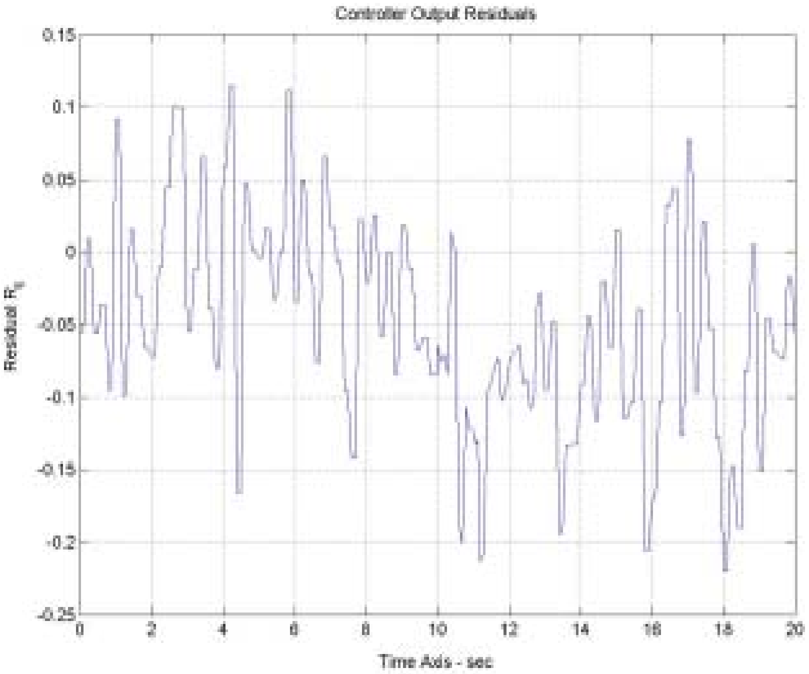
Residuals are collected at both the system output and controller output nodes.
As expected, the controller output residuals are more sensitive (than the system output) to the fault.

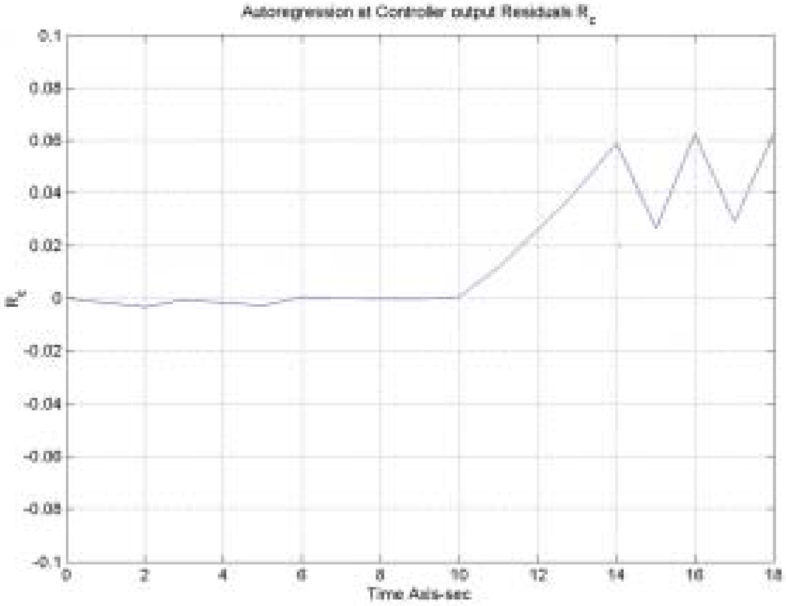
The residuals are modeled with AR-Kalman-filter, and the AR-predicted residual signal shows drastically enhanced performance for early fault detection/warning.

Issues in closed-loop FDI: in the presence of controllers, residuals at the system output becomes less sensitive to faults; the smarter the controller, the worse the output residuals!

Raw Residuals and AR-Predicted Values

(controller output)

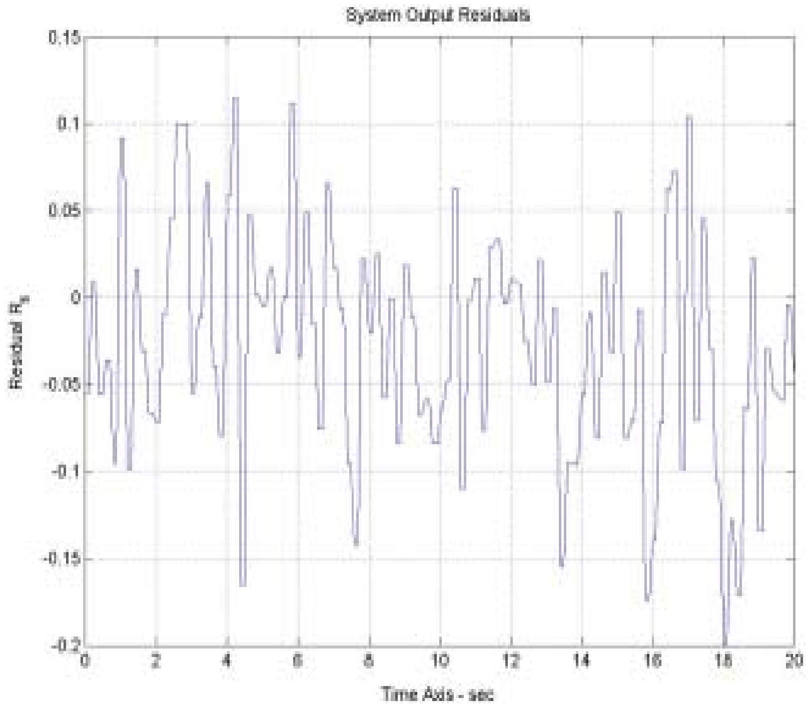


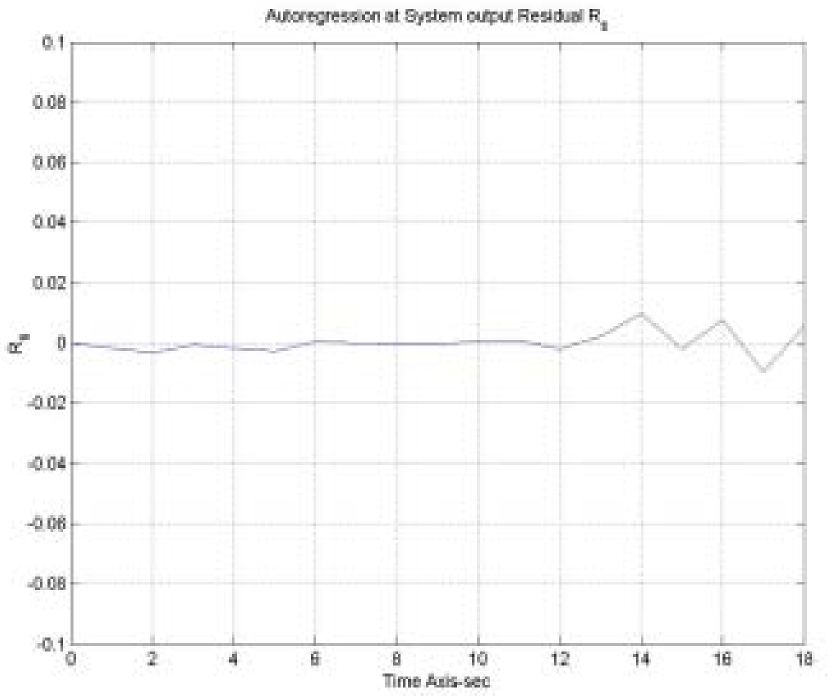


Fault starts at time 12 sec.

Raw Residuals and AR-Predicted Values

(system output)





Fault starts at time 12 sec.

Using the AR-predicted residuals at the controller output appears to be the best option.

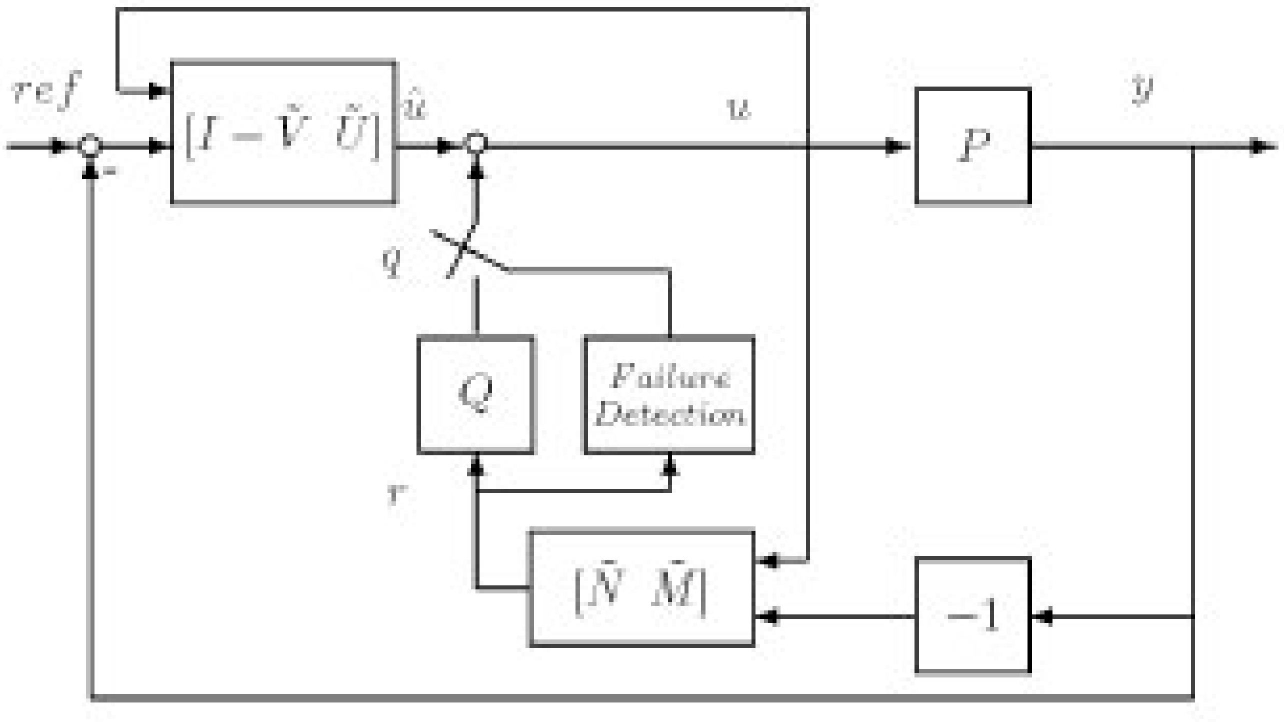
If the residuals are purely random, then any attempt to fit AR or any other type of model must fail – essentially, the Kalman filter will then simply extract the zero signal from the noisy data.
When a fault starts, the AR-parameters will shape up according to the type of the fault. All we need is an excellent real-time AR estimation tool.

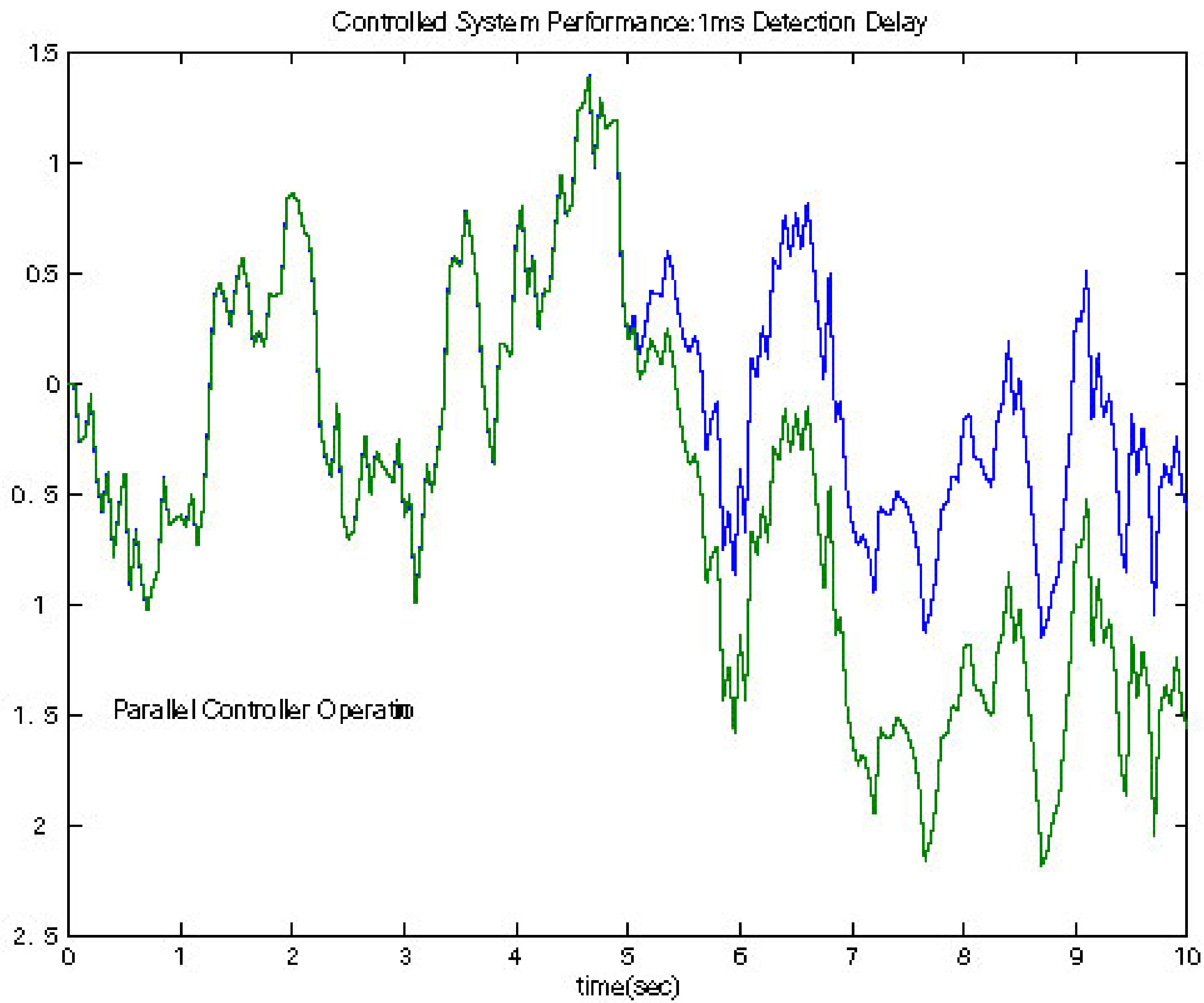
Students:

Sundara Kumar (AR modeling, closed-loop FDI)
Venu Gopal Siddhanti (EKF for TLRN training)
Nageswara Rao (quality of residuals, hypothesis tests)
Dilip Vutukuru, Silpa Mutukuru, Karuna Pilla
(closed-loop FDI, smart controllers and FDI)

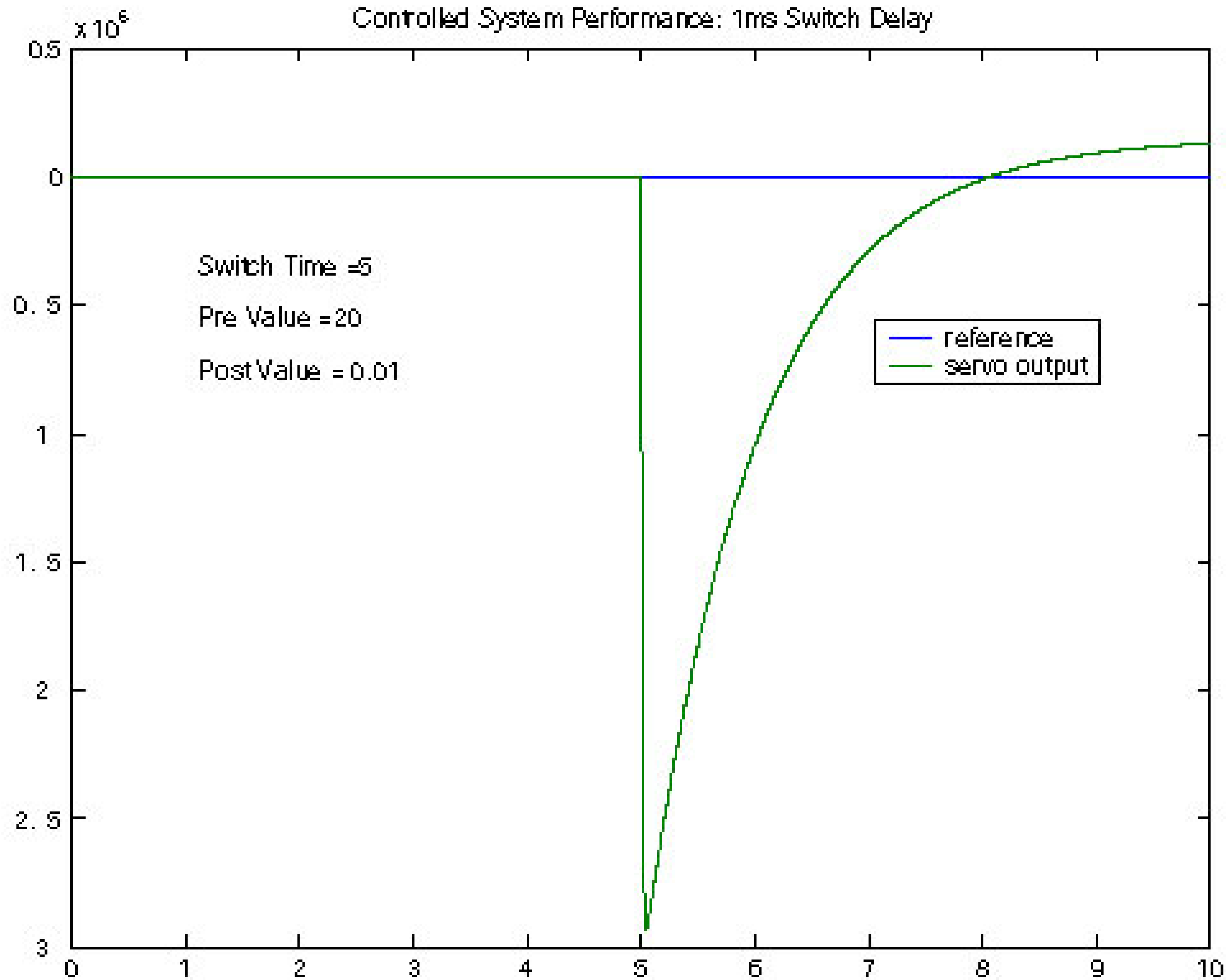
Min Luo (FDI, Subspace signatures)
Pallavi Chetan (STFT signatures, clustering)
Santosh Desiraju (Detection of Change)
James Henderson (Strain test data analysis)

A Curious Little Problem





Controlled System Performance:1ms Detection Delay
1.5
1
0.5
0
0. 5
1
1. 5
2
2. 5
0
1
2
3
4
5
6
7
8
9
10
time(sec)
Parallel Controller Operatio


Controlled System Performance: 1ms Switch Delay
x 10^6
0.5
0
0. 5
1
1. 5
2
2. 5
3
0
1
2
3
4
5
6
7
8
9
10
Switch Time =5
Pre Value =20
Post Value = 0.01
reference
servo output

for a switched system, the difference with respect to the ideal tracking performance corresponds to a combination of free responses from faulted and un-faulted systems and is essentially independent of the controller.