

# F2E5216/TS1002 Adaptive Filtering and Change Detection

Fredrik Gustafsson (LiTH) and Bo Wahlberg (KTH)



Linköpings universitet

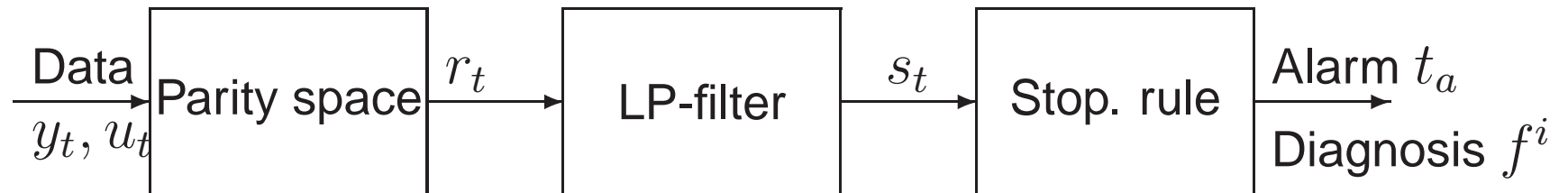


## Lecture 9

### Parity space fault detection

- Deterministic parity space: residual generation, design, examples and sensitivity (Chapter 11)
- Mixed deterministic–stochastic parity space: analysis and design tools (article draft)
- PCA for model-free residual generation

## Fault detection in parity spaces



- A subspace algorithm and variants.
- Alternate approaches.
- Two application examples.

## Data model

State space model:

$$\begin{aligned}x_{t+1} &= A_t x_t + B_{u,t} u_t + B_{d,t} d_t + B_{f,t} f_t \\ y_t &= C_t x_t + D_{u,t} u_t + D_{d,t} d_t + D_{f,t} f_t.\end{aligned}$$

Model over sliding window:

$$Y_t = \mathcal{O} x_{t-L+1} + H_u U_t + H_d D_t + H_f F_t$$

where

$$Y_t = \begin{pmatrix} y_{t-L+1} \\ y_{t-L+2} \\ \vdots \\ y_t \end{pmatrix}, \quad \mathcal{O} = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{L-1} \end{pmatrix}, \quad H = \begin{pmatrix} D & 0 & \cdots & 0 \\ CB & D & & 0 \\ & \ddots & \ddots & 0 \\ CA^{L-2}B & \cdots & CB & D \end{pmatrix}$$

## The algorithm

**Given:** a state space model.

**Design parameters:** sliding window size  $L$  and residual structure  $R$ .

**Compute recursively:**

1. The data vectors  $Y_t$  and  $U_t$  and the model matrices  $\mathcal{O}$ ,  $H_d$ ,  $H_f$ .

2. Let  $\mathcal{N}$  be a any basis for the null space of  $[\mathcal{O}, H_d]$ , so  $\mathcal{N}^T [\mathcal{O}, H_d] = 0$ . In Matlab formalism:

```
[U,D,V]=svd([O Hd]);
```

```
n=rank(D);
```

```
N=U(:,n+1:end);
```

3. Rotate and scale the null space to  $w = TN$ , so the response to unit faults is  $R$ , that is,  $w^T H_f F_t^i = \mathcal{N}^T T^T H_f F_t^i = R_{:,i}$ . In matlab formalism:

```
T = R / (N*Hf*kron(ones(L,1),[f1 f2 f3]));
```

```
w = (T*N)';
```

**4.** Compute the residual  $r = w^T (Y_t - H_u U_t)$ ,

$$r = w' * (Y - H_u * U) ;$$

**5.** Change detection if  $r^T r > 0$ , or  $r^T r > h$  considering model uncertainties.

**6.** Change isolation. Fault  $i$  in direction  $f^i$  where  $i = \arg \max_i r^T R_i$ .  $R_i$  denotes column  $i$  of  $R$ .

$$[ \text{dum} , i ] = \max ( r' * R ) ;$$

Example on  $R$ :

Fault	$f^1$	$f^2$	$f^3$
$r_t^{(1)}$	1	0	0
$r_t^{(2)}$	0	1	0
$r_t^{(3)}$	0	0	1

## Does it work?

When does a parity equation exist? Condition:  $\text{rank}(\mathcal{N}) > 0$ .

$$\text{rank}(\mathcal{N}) = n_y L - \text{rank}(\mathcal{O}) - \text{rank}(H_d)$$

$$\text{rank}(\mathcal{O}) = n_x$$

$$\text{rank}(H_d) = n_d L$$

$$\Rightarrow \text{rank}(\mathcal{N}) = L(n_y - n_d) - n_x$$

Assumption: the column spaces of  $\mathcal{O}$  and  $H_d$  do not overlap

$$\text{rank}(\mathcal{N}) \geq L(n_y - n_d) - n_x.$$

## Comments

- What are the requirements for isolation?
- What is the minimal filter order (the “dead-beat detector”)? Will give minimum delay for detection.

## Alternate approaches

- Most papers assume continuous time!
- An observer can be written as

$$\hat{x}_t = C_y(q)y_t + C_u(q)u_t,$$

The residual can be taken as a linear combination of the state

$$r_t = L\hat{x}_t \triangleq A(q)y_t - B(q)u_t.$$

- An input-output approach

$$y_t = H_u(q)u_t + H_d(q)d_t + H_f(q)f_t$$



A residual generator may be taken as

$$\begin{aligned}
 r_t &= W^T(q) \begin{pmatrix} y_t \\ u_t \end{pmatrix} \\
 &= W^T(q) \underbrace{\begin{pmatrix} H_u(q) & H_d(q) \\ I & 0 \end{pmatrix}}_{M(q)} \begin{pmatrix} u_t \\ d_t \end{pmatrix} + W^T(q) \begin{pmatrix} H_f(q) \\ 0 \end{pmatrix} f_t
 \end{aligned}$$

Design:  $M(q)$  must belong to the left null space of  $W^T(q)$ , a filter of order  $L$ .

## Simulation example: DC motor

$$G(s) = \frac{1}{s(s+1)}$$

sampled with sample interval  $T_s = 0.4\text{s}$ . The state space matrices with  $x^1$  being the angle and  $x^2$  the angular velocity.

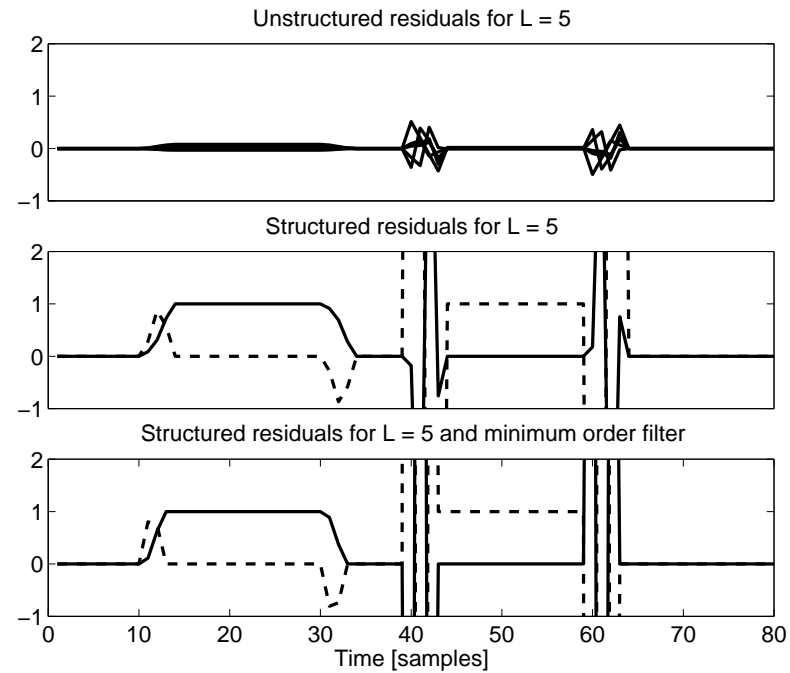
- Matlab demo `lecture('11a')`

## Vertical aircraft dynamics

Inputs	Outputs	States
$u_1$ : spoiler angle [ $0.1\text{deg}$ ] $u_2$ : forward acceleration [ $m/s^2$ ] $u_3$ : elevator angle [ $deg$ ]	$y_1$ : relative altitude [ $m$ ] $y_2$ : forward speed [ $m/s$ ] $y_3$ : pitch angle [ $deg$ ]	$x_1$ : altitude [ $m$ ] $x_2$ : forward speed [ $m/s$ ] $x_3$ : pitch angle [ $deg$ ] $x_4$ : pitch rate [ $deg/s$ ] $x_5$ : vertical speed [ $deg/s$ ]

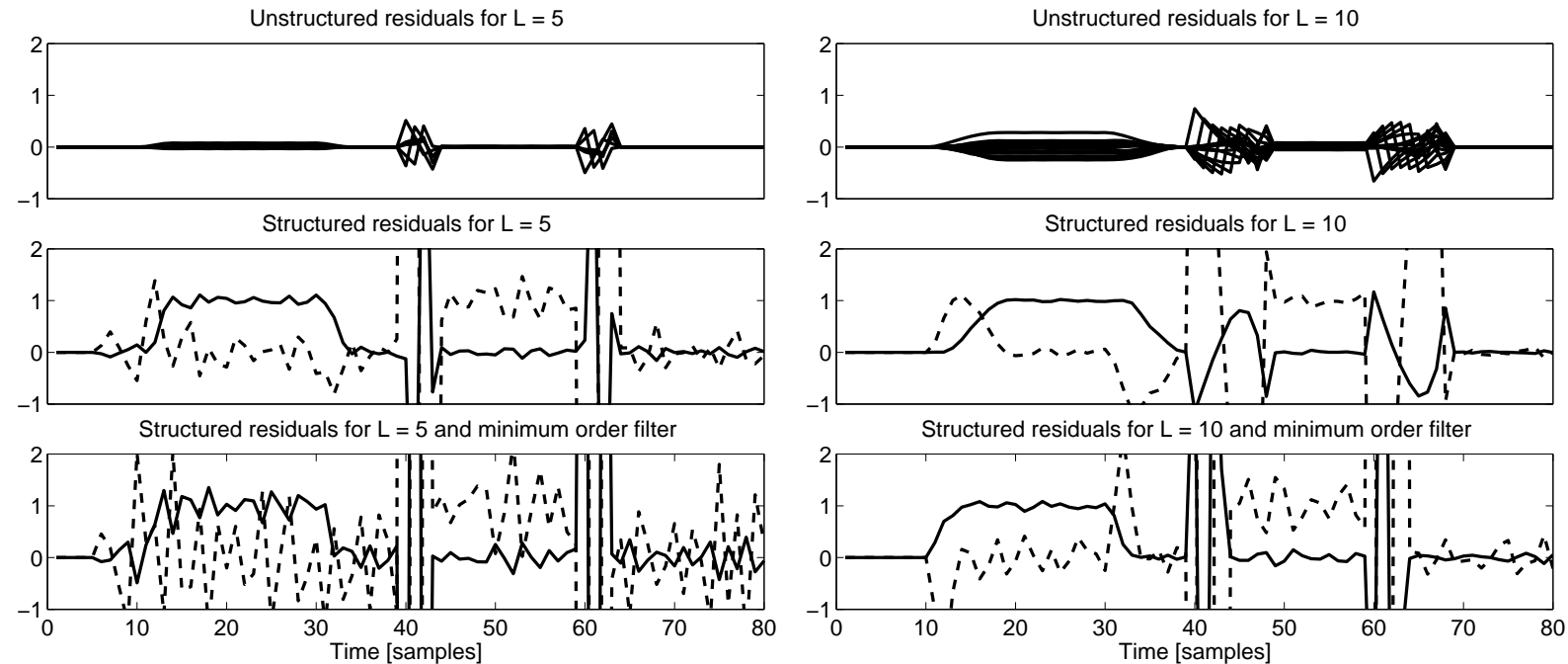
- Matlab demo `lecture('11b')`

# Result



Residuals for sliding window  $L = 5$  for the aircraft model. Upper plot shows unstructured residuals and middle plot structured residuals. Lower plot shows minimal order residuals.

## Result with very small noise



Residuals for sliding window  $L = 5$  and  $L = 10$ , respectively, for the aircraft model. Upper plot shows unstructured residuals and middle plot structured residuals. Lower plot shows minimal order residuals.

Noisy observations

$$r_t = w^T (Y_t + E_t - H_u U_t) = w^T E_t.$$

$E_t$  stacked vector of measurement noises

$$\text{Cov}(E_t) = \mathbb{E}(E_t E_t^T) = I_L \otimes R.$$

The covariance matrix of the residuals is given by

$$\text{Cov}(r_t) = w^T \text{Cov } E_t w = w^T (I_L \otimes R) w.$$

Note: the algorithm produces a  $w$  where the components of  $w$  differ in order of magnitudes.

## Mixed stochastic–deterministic model

*Linear state space model.*

$$\begin{aligned}x_{t+1} &= A_t x_t + B_{u,t} u_t + B_{d,t} d_t + B_{f,t} f_t + B_{v,t} v_t \\ y_t &= C_t x_t + D_{u,t} u_t + D_{d,t} d_t + D_{f,t} f_t + e_t.\end{aligned}$$

- Deterministic known input  $u_t$ .
- Deterministic unknown disturbance  $d_t$ .
- Deterministic unknown *additive* fault input  $f_t = m_t f^i$  ( $f^i$  unit vector  $i$ ).
- Stochastic unknown state disturbance  $v_t$  and measurement noise  $e_t$ .

- *Residual* as FIR filter of observations (length  $L$ )

$$\begin{aligned} r_t &= w^T (Y_t - H_u U_t) \\ &= w^T (\mathcal{O}x_{t-L+1} + H_d D_t + H_f F_t + H_v V_t + E_t) \\ &= w^T (H_f F_t + H_v V_t + E_t). \end{aligned}$$

- *Parity space* defined by  $w^T \mathcal{O} = 0$  and  $w^T H_d = 0$ .
- The Kalman filter approach, neglecting disturbance  $D_t$ , estimating  $x_{t-L+1}$  using the minimum variance principle, plugging in the estimate above, also ensures  $w^T \mathcal{O} = 0$ ! *Parity space and Kalman filter are dual approaches.*
- With  $w$  defined as above, the residual follows

$$r_t = \text{N}(H_f F_t, H_v \text{Cov}(V_t) H_v^T + \text{Cov}(E_t)).$$



## DC motor example

A DC motor

$$G(s) = \frac{1}{s(s+1)} = \frac{1}{s^2 + s},$$

sampled with 0.4 second, facilitates the geometrical interpretation.

- Faults:** Torque and sensor offsets.
- Measurements:** Angle and angular velocity.
- Noise:** Measurement and torque noise.
- Disturbances:** None.

## Background

Traditionally, either a stochastic ( $d_t = 0$ ) or a deterministic ( $v_t = 0, e_t = 0$ ) framework is analysed in the literature, with a few exceptions.

Gertler, 1998:

“It is possible to design a matrix filter which whitens the complete residual vector, this procedure however is quite complex and, to our understanding, does not allow for the preservation of the geometric residual properties. Therefore, here we will restrict ourselves to the scalar whitening problem.”

## Suitable basis for residuals

Any non-singular transformation can be applied to the residual  $r_t = w^T (Y_t - H_u U_t)$ .

**Alternative 1.** Structured residual as a basis in the algebraic approach. Examples:

Res \ Fault	$f^1$	$f^2$	$f^3$	or	Res \ Fault	$f^1$	$f^2$	$f^3$
$r^1$	1	0	0		$r^1$	0	1	1
$r^2$	0	1	0		$r^2$	1	0	1
$r^3$	0	0	1		$r^3$	1	1	0

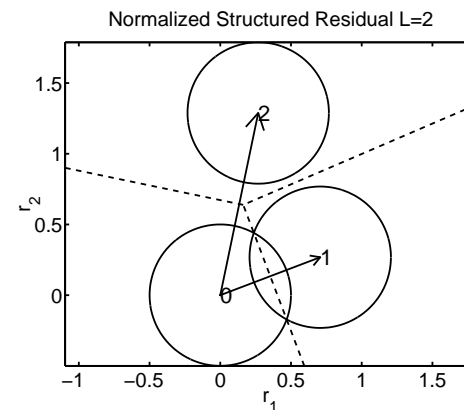
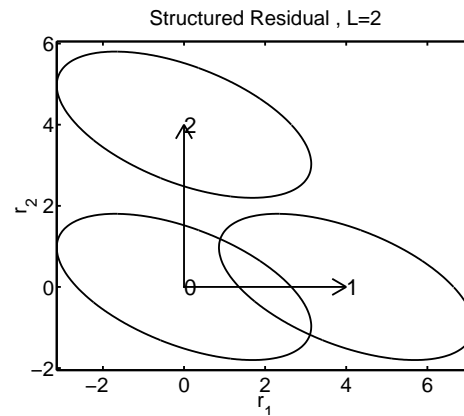
**Alternative 2.** Rotate and scale residuals to get a  $\text{Cov}(r_t) = I$ .  
 Advantage: facilitate multidimensional integration to calculate probabilities.

# Stochastic normalization of residual basis

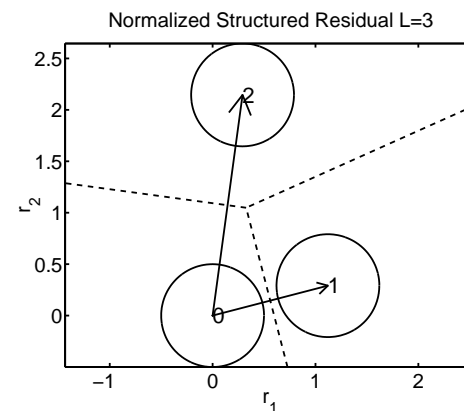
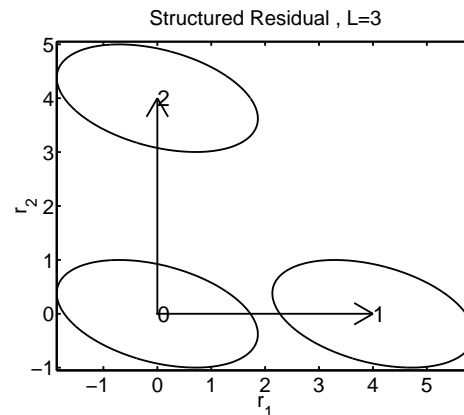
Structured residual

Normalized residual

$L = 2$ :



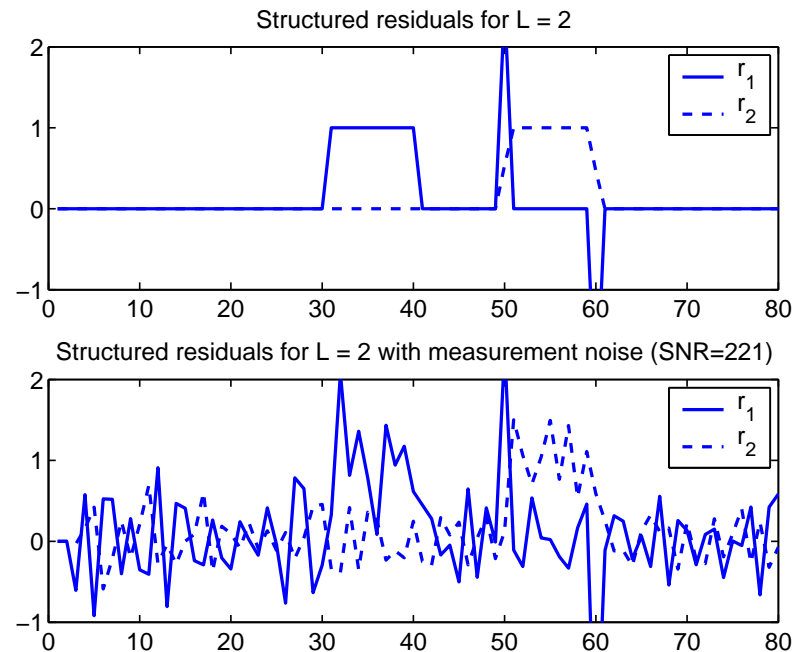
$L = 3$ :



## Motivation 1 for stochastic normalization

Gives simple geometric interpretation of which faults are easy to detect and diagnose, and which will give quite similar residuals.

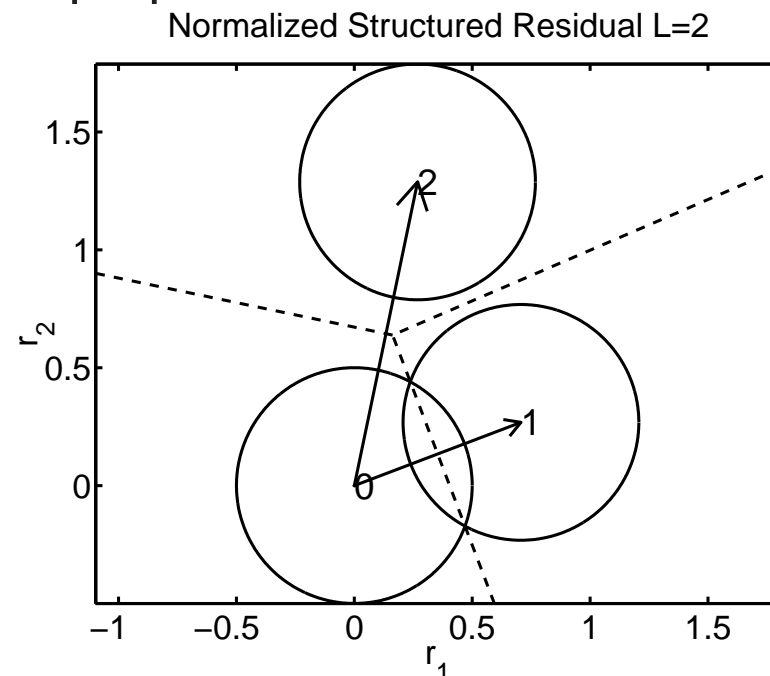
Example: Response in structured residual as a function of time for both faults, without measurement noise and with (small) measurement noise.



## Motivation 2 for stochastic normalization

Enables evaluation of probability for incorrect diagnosis. Similar to the Gaussian vector model in telecommunication for computing probabilities for incorrect symbol detection.

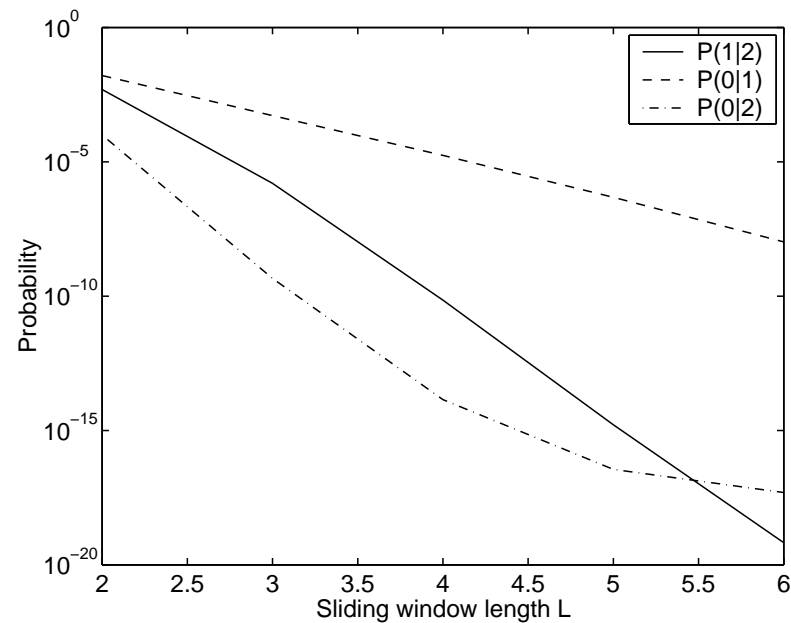
Idea: approximate the multi-dimensional integral with a one-dimensional one perpendicular to each decision line



## Example: algorithm design

**Question:** How does the detection and diagnosis design parameter  $L$  affect incorrect decisions??

**Answer:** For a typical fault magnitude, the miss-classification probabilities change as follows:



## Example: system design

**Question:** What about replacing the expensive velocity sensor with a cheaper one, increasing the noise variance from 1 to 10??

**Answer:** For a typical fault magnitude and  $L = 2$ , the diagnosis matrix changes as

$$P^{(1:2,1;2)} = \begin{pmatrix} 0.995 & 0.005 \\ 0.005 & 0.995 \end{pmatrix} \Rightarrow P^{(1:2,1;2)} = \begin{pmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{pmatrix}.$$

**Action:** increase  $L$  and accept longer delay for detection.



## Conclusions on stochastic parity spaces

**Idea:** Mixed deterministic fault/disturbance and stochastic process/measurement noise framework, that extends KF and parity space approaches.

**Results:** Explicit analytical formulas (no Monte Carlo simulations) for probabilities of incorrect diagnosis. Besides geometrical insights, it provides an interactive tool for system and algorithm design, and sensitivity analysis.

**Assumptions and limitations:**

- Additive faults (multiplicative faults harder to analyze!)
- Linear system (EKF-like approximations otherwise)
- Known model (but a model-free PCA approach exists)
- Gaussian noise (but central limit theorem often applies anyway)
- FIR residuals (but fairly easy to generalize to IIR residuals)

## PCA: A data-based approach

PCA common approach in chemical engineering. Typically, they have thousands of measurements and no obvious causality (data  $z_t$  cannot be split into input  $u_t$  and output  $y_t$ ).

Idea: Split data into model and residual

$$Z_t = \begin{pmatrix} Y_t \\ U_t \end{pmatrix} = \hat{Z}_t + \tilde{Z}_t = P_x x_t + P_r r_t,$$

and compute recursively the residual

$$\begin{aligned} r_t &= P_r^T Z_t \\ \text{Cov}(r_t) &= D_r \\ \bar{r}_t &= D_r^{-1/2} P_r^T Z_t. \end{aligned}$$

**Algorithm:**

1. Covariance matrix estimation and SVD.

$$\hat{R}_Z = \frac{1}{N-L} \sum_{t=L+1}^N Z_t Z_t^T = P D P^T.$$

where  $P^T P = P P^T = I$ .

2. Split the SVD into two parts:

$$P = \begin{pmatrix} P_x & P_r \end{pmatrix}, \quad D = \begin{pmatrix} D_x & 0 \\ 0 & D_r \end{pmatrix}$$

3. Model and residuals are now computed as

$$\begin{aligned} \hat{Z}_t &= P_x P_x^T Z_t \\ \tilde{Z}_t &= P_r P_r^T Z_t. \end{aligned}$$