FAULT DETECTION AND ISOLATION: AN OVERVIEW

María Jesús de la Fuente
Dpto. Ingeniería de Sistemas y Automática
Escuela de Ingenierías Industriales
Universidad de Valladolid

Outline

- Introduction.
- Systems and faults:
  - What is a fault
  - Fault types
- Characteristics of FDI methods
- Diagnosis approaches
  - Model – based methods
  - Model – free methods
- Data driven methods
  - Application of data driven methods to whole plants

Industrial Processes Automation 1

- Many advances in Control Engineering but:
  - Systems do not render the services they were designed for
  - Systems run out of control
  - Energy and material waste, loss of production, damage the environment, loss of humans lives

Industrial Processes Automation 2

- Malfunction causes:
  - Design errors, implementation errors, human operator errors, wear, aging, environmental aggressions

Fault diagnosis

Safety Levels

Detection
Isolation
Identification

$

Fault Tolerant Control

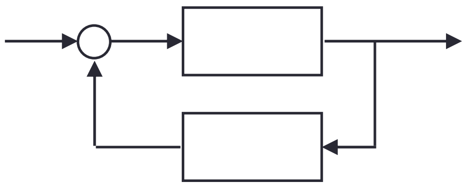
Predictive Maintenance

Industrial Processes Automation 3

- **Fault diagnosis**:
  - Fault detection: Detect malfunctions in real time, as soon and as surely as possible
  - Fault isolation: Find the root cause, by isolating the system component(s) whose operation mode is not nominal
  - Fault identification: to estimate the size and type or nature of the fault.
- **Fault Tolerance**:
  - Provide the system with the hardware architecture and software mechanisms which will allow, if possible to achieve a given objective not only in normal operation, but also in given fault situations

Industrial Processes Automation and 4

Automatic
control



FDI
scheme

Safety Levels

- Detection
- Isolation
- Identification

$

Fault Tolerant
Control

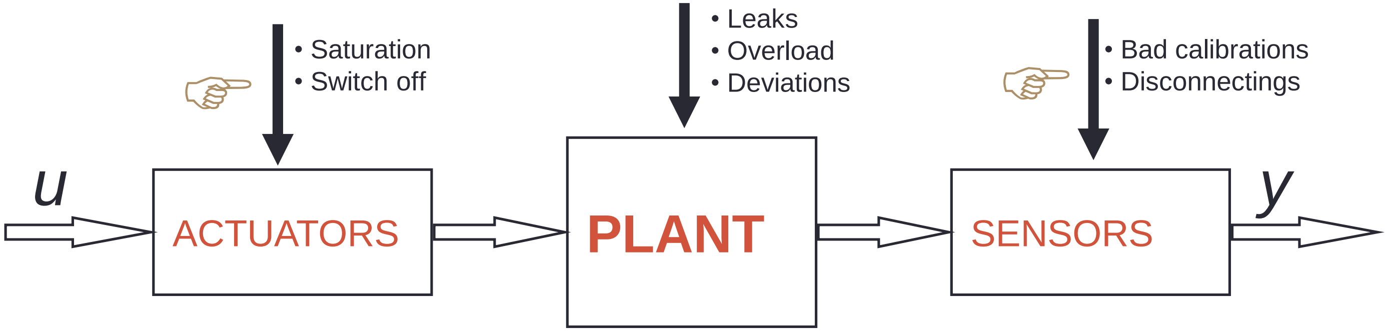
Predictive
Maintenance

Fault concepts

- **Fault**: an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition.
  - Causes: design errors, implementation errors, human errors, use, wear, deterioration, damages, ageing…
  - Consequences: worse performances, energy waste, waste of raw materials, economic losses, lower quality, lower production, environmental damages, human damages…

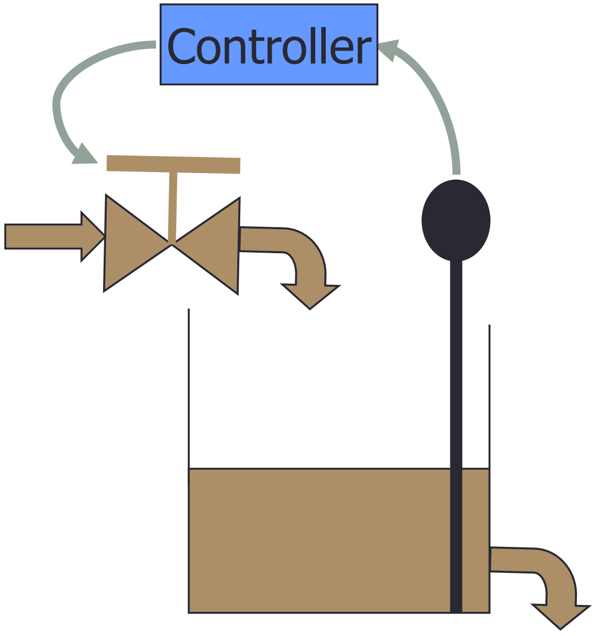
Fault types

- Depending on the magnitude of the fault:
  - Acceptable departure from the usual state.
  - Fault.
  - Failure. Catastrophic. Permanent interruption of a system's ability to perform a required function under specific operating conditions.
- Depending on the localization of the fault:
  - External fault: interactions between system and environment are not compatible with goals.
  - Internal fault. Depending on the faulty component: system, sensor, actuator



Example

- Internal faults
  - Process: Tank leakage, clogged pipe
  - Sensor: Offset
  - Actuator: Valve is blocked
- External faults:
  - inflow is too small,
  - input valve totally open,
  - level below setpoint

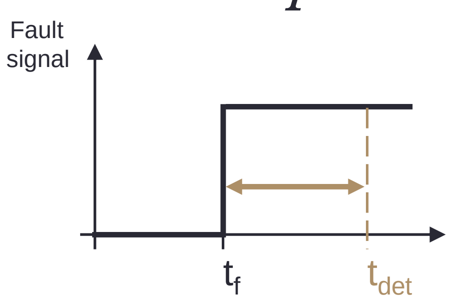


Faults type

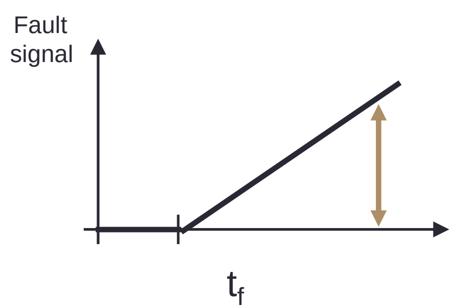
- Depending on the temporal aspects
  - Abrupt fault: sudden and considerable. Model: step. Example: offset
  - Incipient or evolutive fault: affects slowly. Model: ramp, exponential, parabola. Example: drift
  - Intermittent fault. Model: pulses

How ?

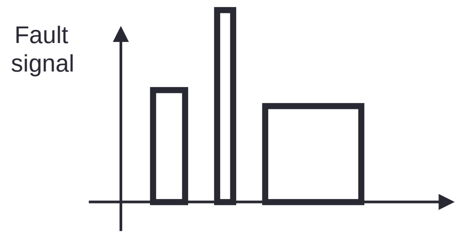
Abrupt



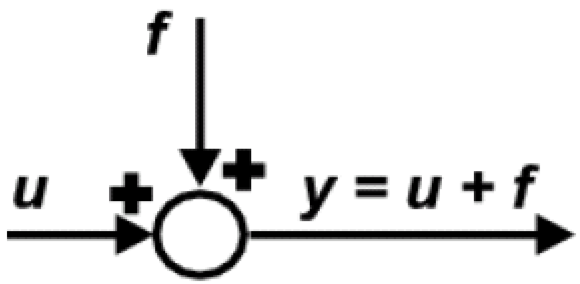
Evolutive



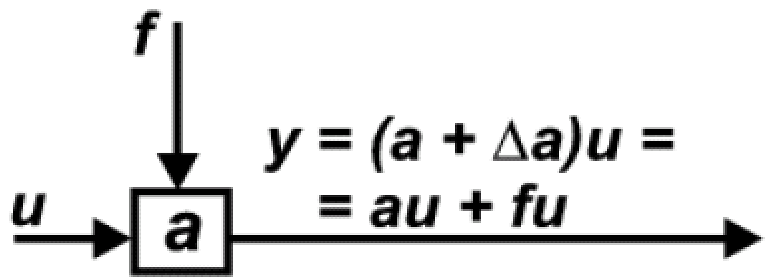
Intermittent



Faults types



- Additive fault: fault=f

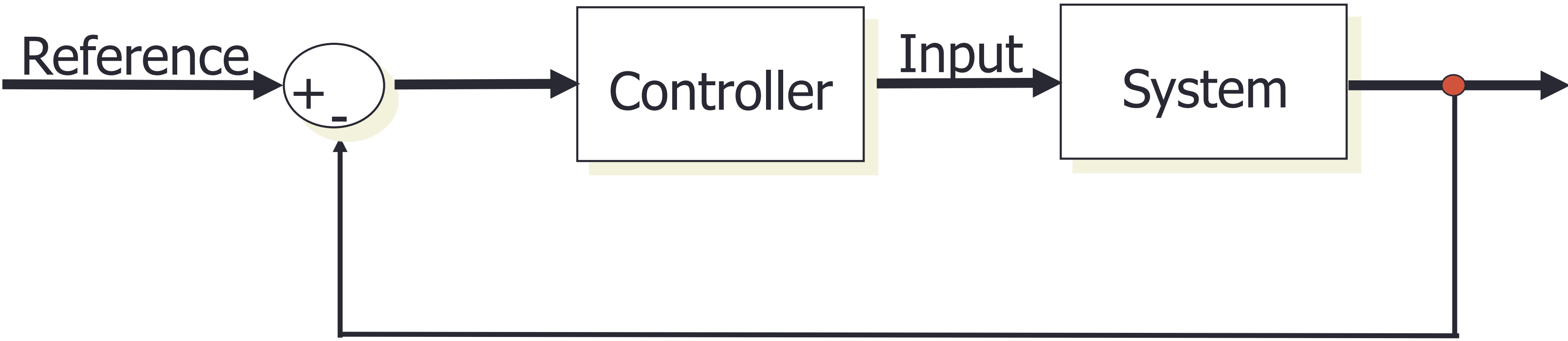


- Multiplicative fault: fault = Δa

- Depending on the way the faults affect to the behaviour of the system
  - Additive fault. Changes at output depend of magnitude of the fault and do not depend of inputs: offsets in sensors and actuators, disturbances
  - Multiplicative fault. Changes at output depend of the magnitude of the fault and of inputs: gain of a sensor, deterioration, corrosion, erosion, loss of energy…

Fault tolerant control (FTC)

- Is intended to continue the system operation as long as possible in the presence of one or several faults, provided both efficiency and security remain acceptable
- The aims at making the system stable and retain acceptable performance under faults.



Fault tolerant control (FTC)

- Techniques depending on the size of the fault
  - Passive: Robustness (robust control). Single controller performing well even if there are small differences
  - Active:
    - Adaptation (adaptive control). Controller tunes automatically to adapt to bigger differences
    - Fault handling
      - Normal operation: Reconfiguration of the system, accommodation to fault
      - Degraded operation. Change of goals
      - Safe stop

Tasks

- Monitoring. Surveillance of the process.
- Supervision. Surveillance of the process and proposal of solutions (fault handling).

Characteristics of the FDI methods

- False alarms: A fault detected when there is not occurred a fault in the system. It is necessary a low rate of false alarms
- Missed detection: A fault that occurs and it is not detected
- Detection time: (delay in the detection). Fault must be detected as soon as possible
- Isolation errors: distinguish a particular fault from others
- Sensibility: the size of fault to be detected
- Robustness: (in terms of uncertainties, models mismatch, disturbances, noise ,....)

Characteristics of the FDI methods

Confusion matrix	Fault	No fault
Detection	*Ok* *True positive*	*False alarm* *False positive*
No detection	*Missed alarm* *False negative*	*Ok* *True negative*

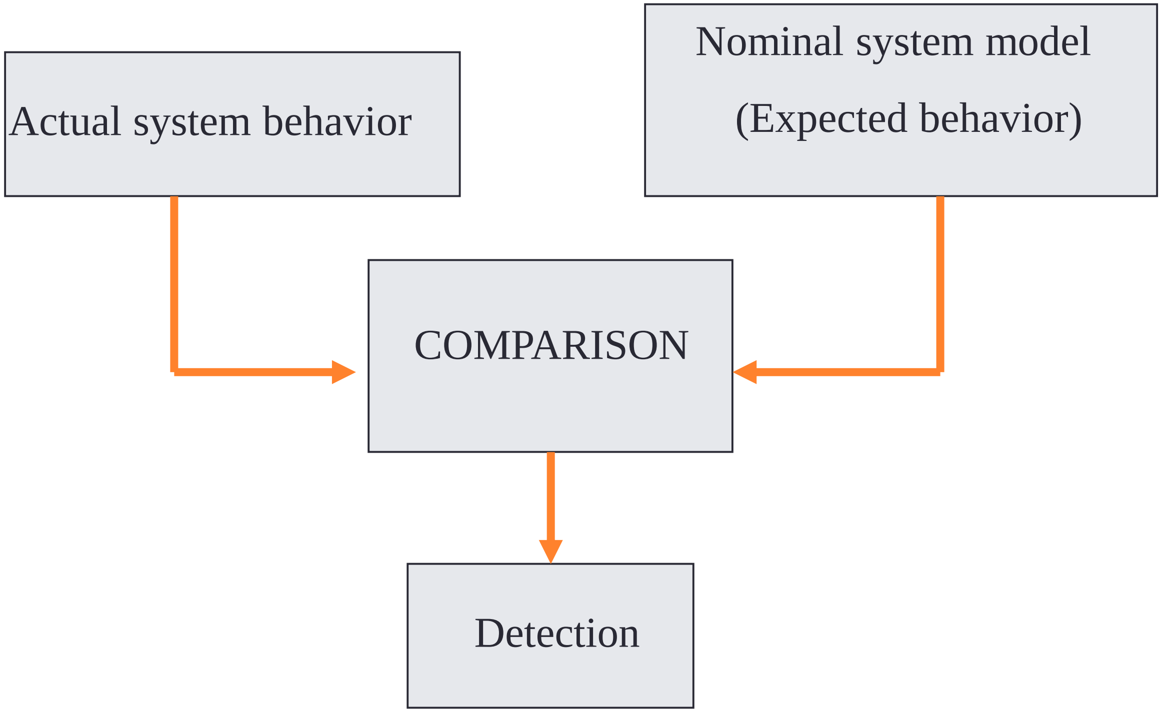
- Detection errors: reliability => false and missed alarms
- Sensitivity: Detection/Fault = TP/ (TP+FN)
- Specifity: No Detection / No Fault = TN / (TN+FP)
- False positive rate: Detection/No fault = FP / (FP+TN)
- False negative rate: No detection / Fault = FN /(FN+TP)
- Goals:
  - Sensitivity = specifity =1
  - False positive rate = False negative rate =0

FDI: FAULT DETECTION AND ISOLATION METHODS

- FDI methods (Gertler, 1998):
  - model based methods
  - model free methods (methods based on data)

Model based FDI methods

- **Model based approaches**:
  - Analytical redundancy
  - Compare actual system with a nominal model system

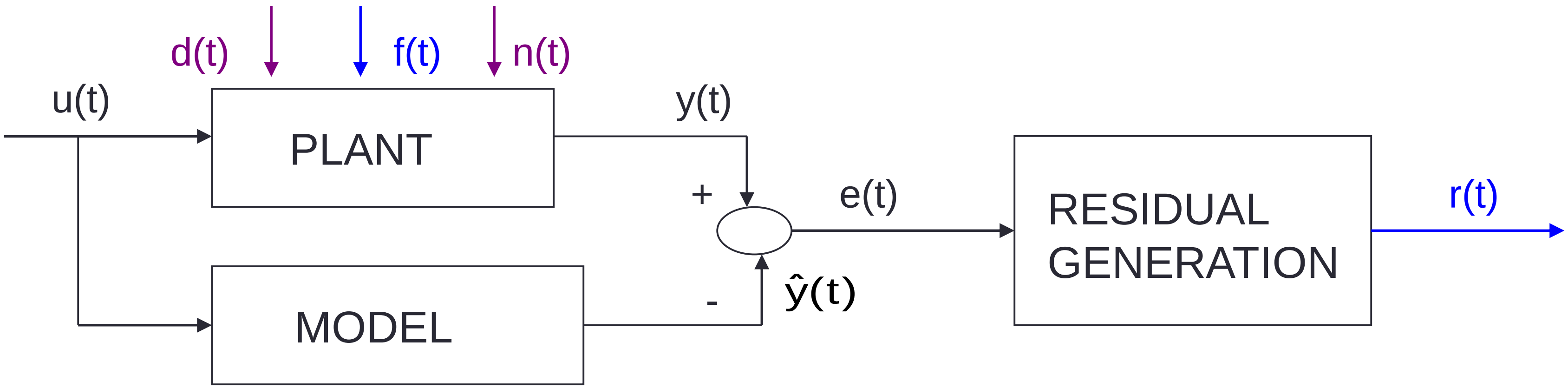


Model based FDI methods

- Model based approaches: two main areas:
  - FDI => from the control engineering point of view
  - DX => Artificial Intelligence point of view
- From FDI:
  - Models:
    - Observers (Luenberger, unknown input etc.)
    - Kalman filters
    - parity equations
    - parameter estimation (Identification algorithms)
    - Structural analysis: ARR: analytical redundancy realtions
    - Extension to non linear systems (non-linear models)

Model based FDI methods

- Primary residual: $e(t) = y(t) - \hat{y}(t)$
- r(t) => processed residual



Model based FDI methods

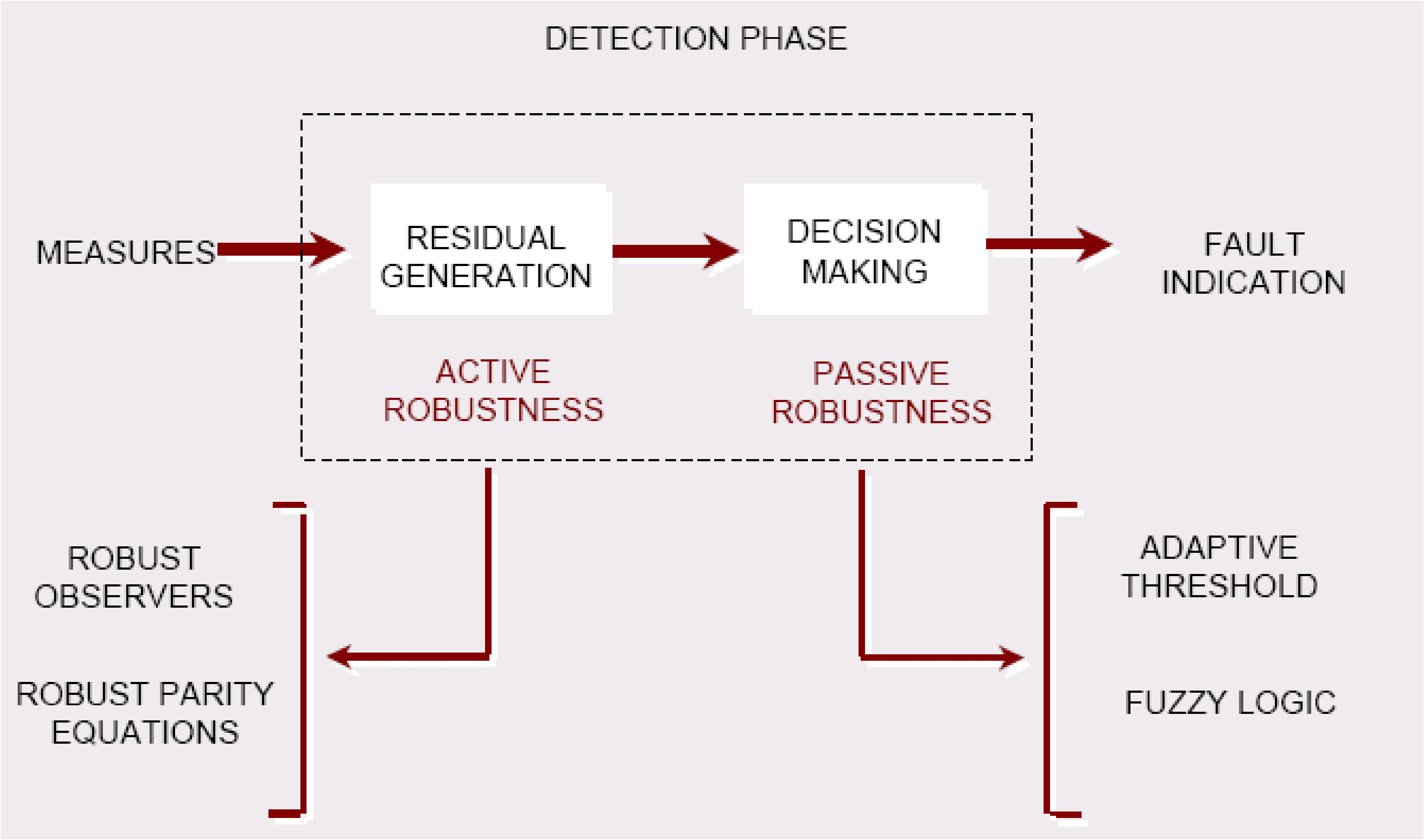
- **Models**: are the output identical to the real measurement?
  - Construct the residuals: $r_t = y_t - \hat{y}_t$
  - Test whether they are zero (true if logic) or not

Problem: $$r_t = \Phi(u_t, y_t, \varepsilon_t, v_t, d)$$

noise disturbances uncertainties

Robust residual generation
or robust residual evaluation

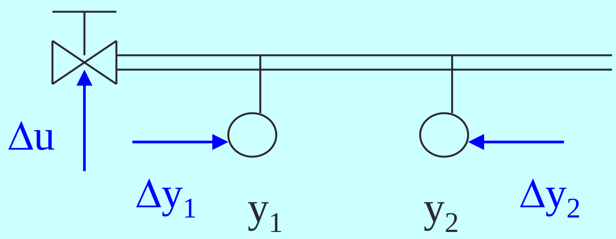
Model based FDI methods



Model based FDI methods

- Fault detectability: to define residuals that are affected by the faults, i.e., residuals that permit to detect faults
  - Residual generation
  - Residual evaluation: several approaches
    - Comparison of the residue with a threshold fixed or an adaptive one
    - Hypothesis testing: SPRT, GLR
- Fault isolability: provide the residuals with characteristic properties that permits to isolate the different faults, i.e., the residuals are built such that each one is associated with one fault (one subset of faults)
  - Directional residues
  - Structured residues

Example



Model:	Model with faults:
$y_1 = f(u)$	$y_1 = f(u) + \Delta u + \Delta y_1$
$y_2 = f(u)$	$y_2 = f(u) + \Delta u + \Delta y_2$

- Primary residuals:

$$e_1 = y_1 - f(u) = \Delta u + \Delta y_1$$

$$e_2 = \underbrace{y_2 - f(u)}_{\text{Computation form}} = \underbrace{\Delta u + \Delta y_2}_{\text{Internal form}}$$

	Δu	Δy_1	Δy_2
r_1	1	1	0
r_2	1	0	1
r_3	0	1	1

Structured residuals

- Processed residuals:

$$r_1 = e_1 = \Delta u + \Delta y_1$$

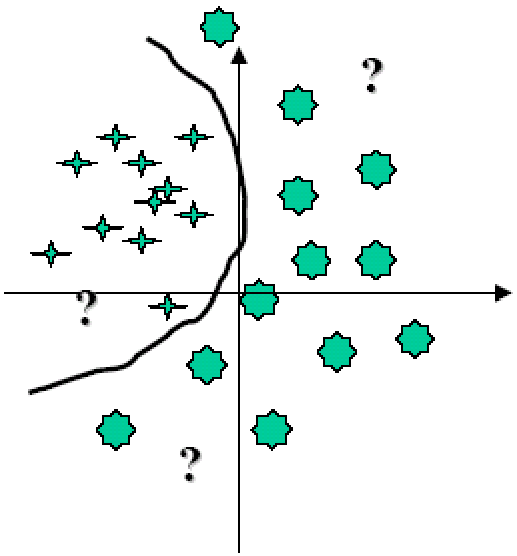
$$r_2 = e_2 = \Delta u + \Delta y_2$$

$$r_3 = e_1 - e_2 = \Delta y_1 - \Delta y_2$$

- Incidence Matrix: dependence between a fault (column) and a residual (row) => 1
- Coincidence between the experimental and theoretical incidence matrix

Data driven methods. Motivation

Data driven methods
Process History Based Methods
Data Mining Methods
Instance Based Methods



- Only experimental data are exploited
- Are indicated for FDI of process when:
  - Mathematical models do not exist or they are incomplete or imprecise
  - Dimensionality (number of variables) or complexity (distributes, non lineal, variant systems) makes unfeasible other techniques
  - There exit or is feasible to get a case base (examples) of previously documented experiences to infer a model

Data driven methods. Tasks

- Preprocessing:
  - Filtering
  - Eliminate outliers, corrupted data.
  - Impute missing data, etc
- Exploratory data analysis:
  - Which are the most significant variables or have all they the same importance?
  - Are the variables redundant?
- Transformation and feature extraction
  - Extract information from raw data or transform the data to get a better representation
- Model construction and validation:
  - Are assumptions made on available data true?. How the representative is the available data (coverage)?. Is the model consistent with the actual data? And with the future?
- Model exploitation:
  - Fault detection and diagnosis

Goal oriented tasks

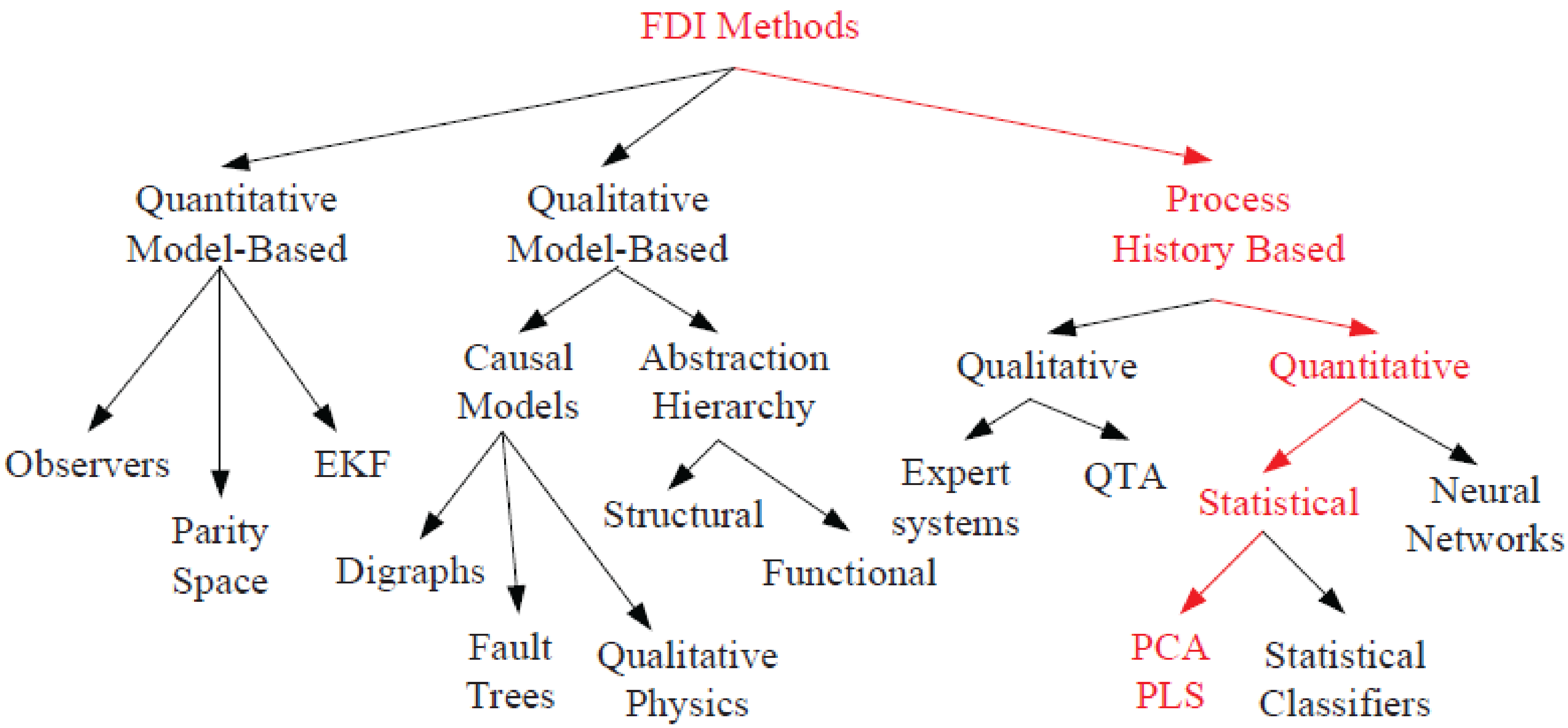
Goal

Refine

Data driven methods

- Computational models: those obtained from methods developed in the area of computer science or AI
  - Clustering methods: classification methods.
  - Decision trees
  - Neural networks
  - Support Vector Machine (SVM)
  - Distance / similarity based methods …
- Statistical models: a probabilistic behavior is assumed in data
  - Parametric models: a predefined function specified by a set of parameters is assumed as a model: distribution function, regressive models, SPC, etc
  - Non parametric models: data correspond with a distribution function but this is neither predefined or parametrized: histograms

Another classification

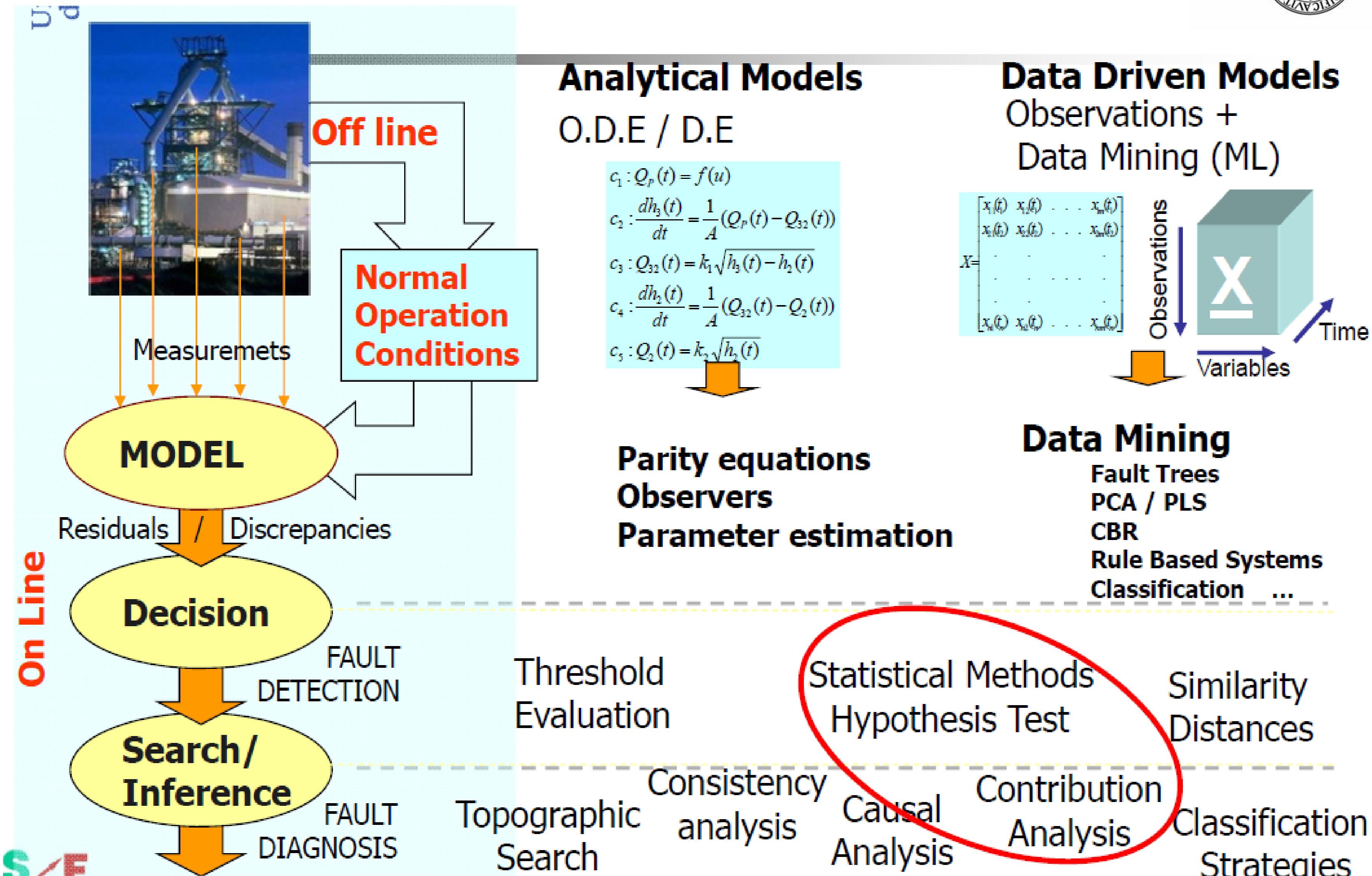


V. Venkatasubramanian et al. / Computers and Chemical Engineering 27 (2003) 293–311

APPLICATIONS

- Data driven methods:
  - Evaporation station of a sugar factory
  - Desalination plant
  - Wastewater treatment plant
  - Water distribution networks

Process monitoring: a global overview



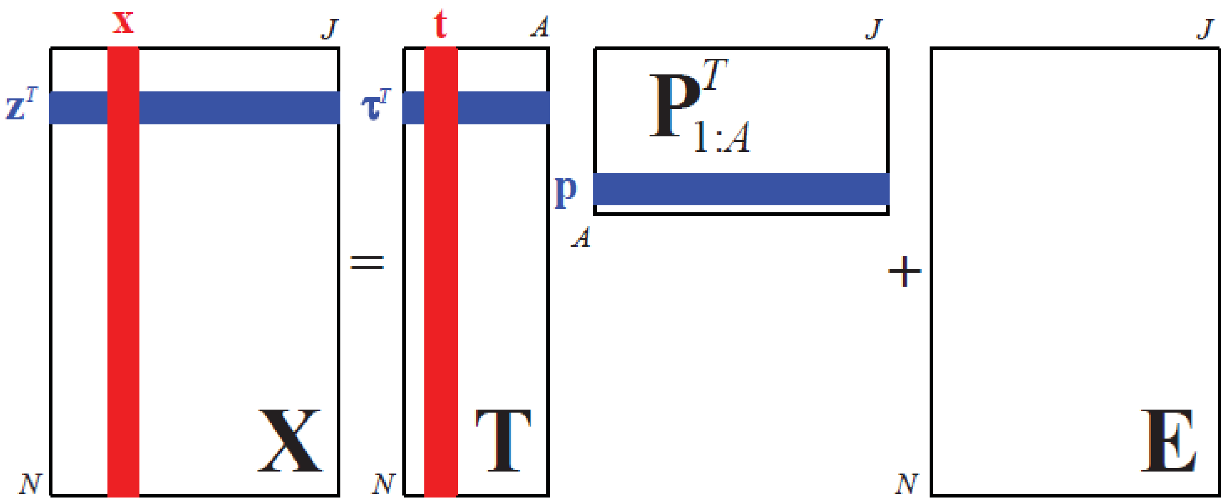
Data driven Methods: PCA

- **PCA** (principal component analysis) is a projection technique that produces a lower dimensional representation:
  - Data is projected onto a space with lower dimension than the original one.
  - Preserves the correlation structure between process variables
  - It is optimal in terms of capturing the variability in the data
- PCA allows to separate into different subspaces the trends of process and noise.
- The PCA structure can be useful in identifying either the variable responsible for the fault and/or the variables most affected by the fault.

Data driven Methods: PCA

The PCA formulation can be expressed as:

$$\mathbf{X} = \mathbf{T}\mathbf{P}_{1:A}^T + \mathbf{E}$$

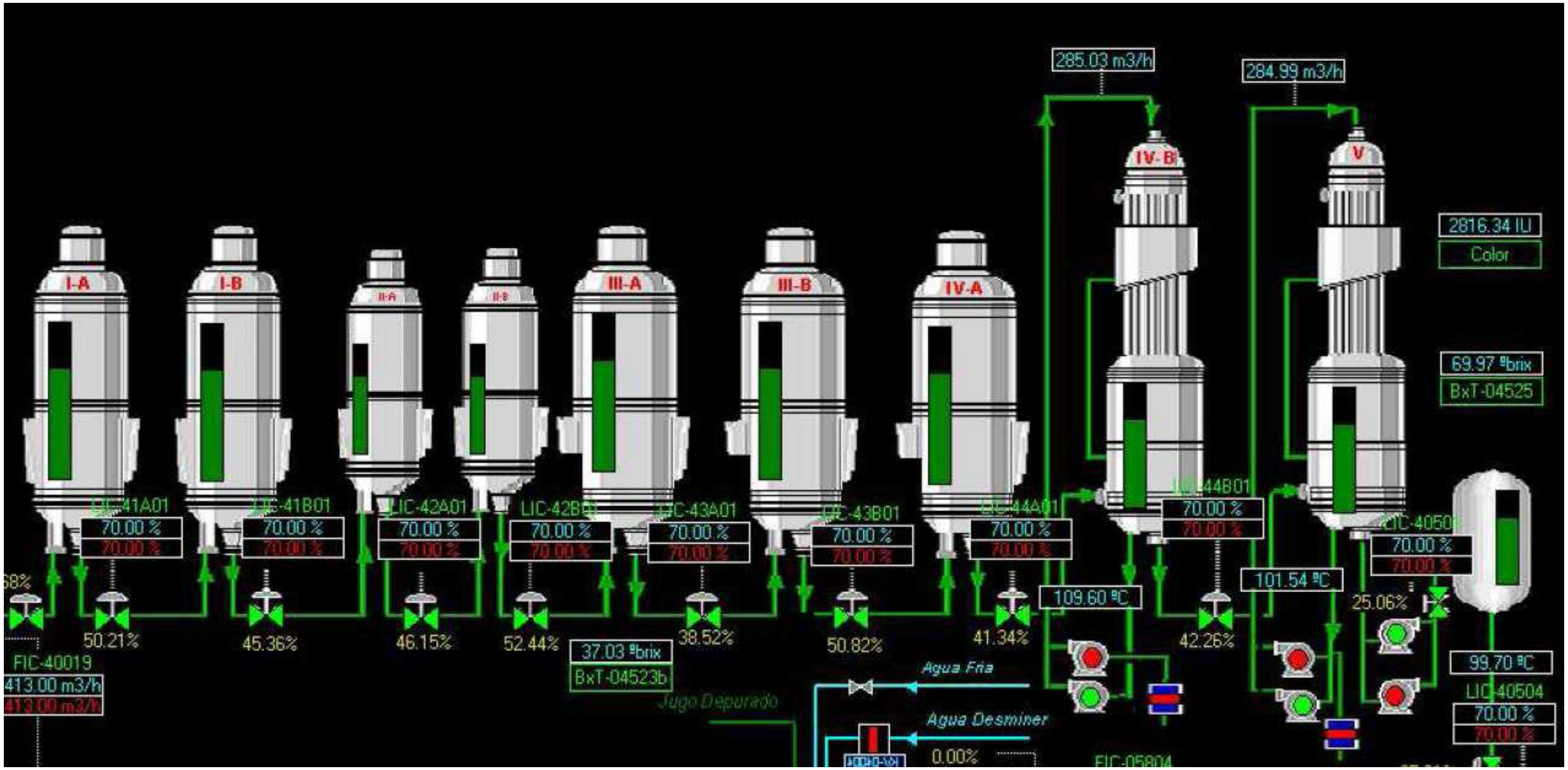


The PCA model can be calculated using the non-linear iterative partial least squares (NIPALS) algorithm or the SVD decomposition.

Data driven Methods: PCA

- To detect faults two statistical are used:
  - **Hotelling's T2** statistic will be used in the **A**-dimension space (A < m number of principal components) to detect misbehaviors based on threshold trespassing.
  - **The Q statistic** will be used to monitor the portion of observation not corresponding to the **m-A** smallest singular values
- To diagnosis the faults:
  - **Contribution plots**: gives an idea of which variable/s in the original space are responsible of the detected fault.

Examples. Evaporation Station



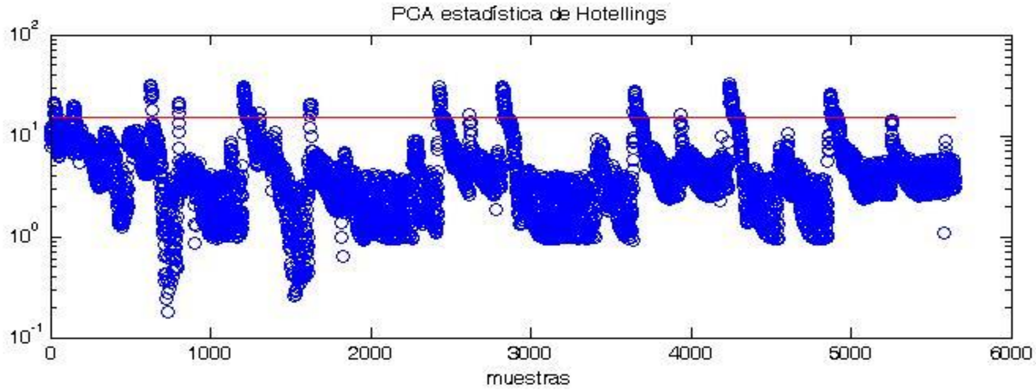
- A very exhaustive first principles model of the system is used to detect the faults, it contains 2,546 equations and 3,699 variables so the faulty behavior can be simulated perfectly.

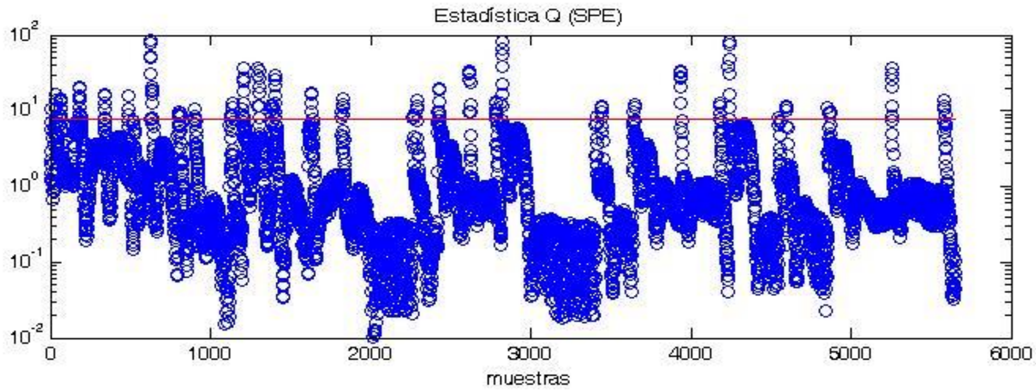
Examples. Evaporation Station

- The faults considered in this system are:
  - Fault 1 (F1). Decay of the performance in one of the evaporators.
  - Fault 2 (F2). Blockage in a valve.
  - Fault 3 (F3). Accumulation of non condensing materials in one of the evaporators.
  - Fault 4 (F4). Sensor offset.
- The variables collected to perform the PCA model are 46 signals of the typical sensors (flows, pressures, temperatures, etc)

Examples. Evaporation Station

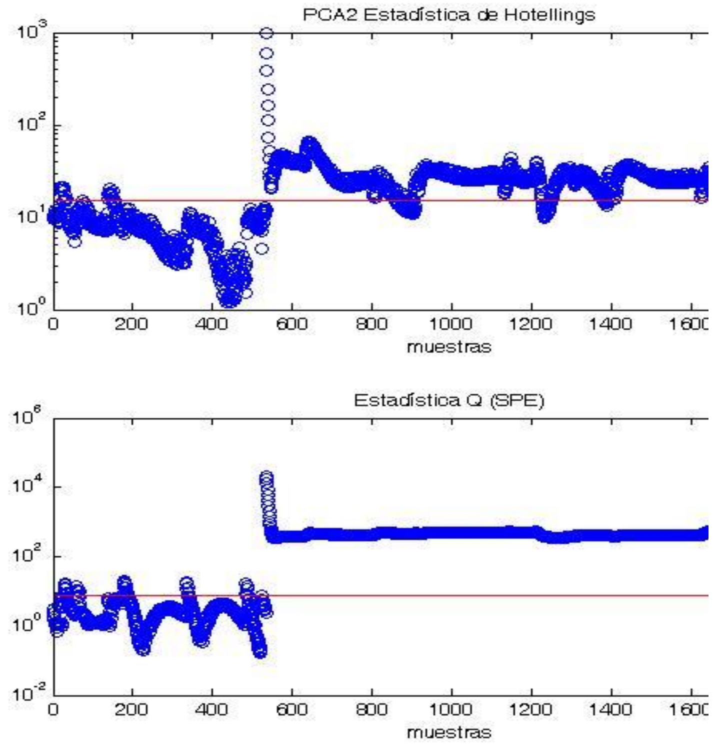
- 5 principal components are obtained, which explain the 95% of the variability of the process.



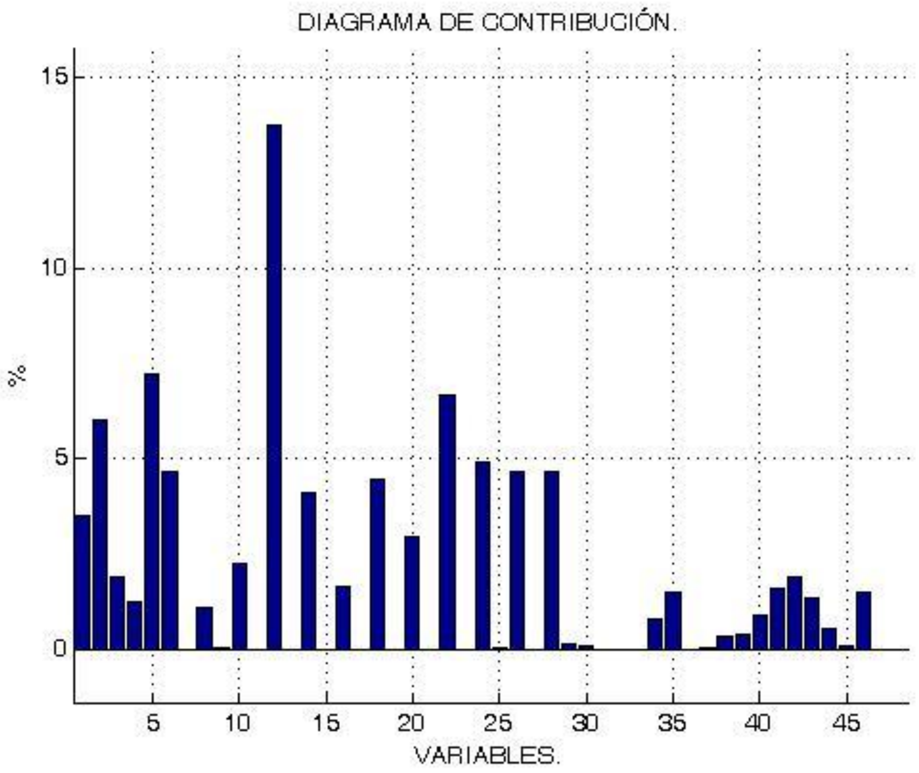


Examples. Evaporation Station

- Fault 1.



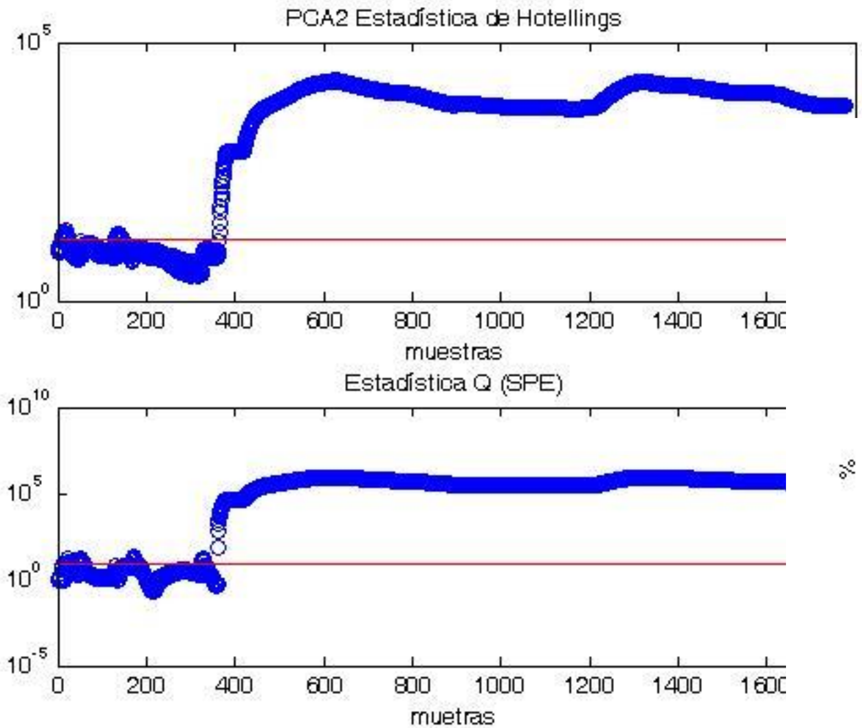
- Contribution plot

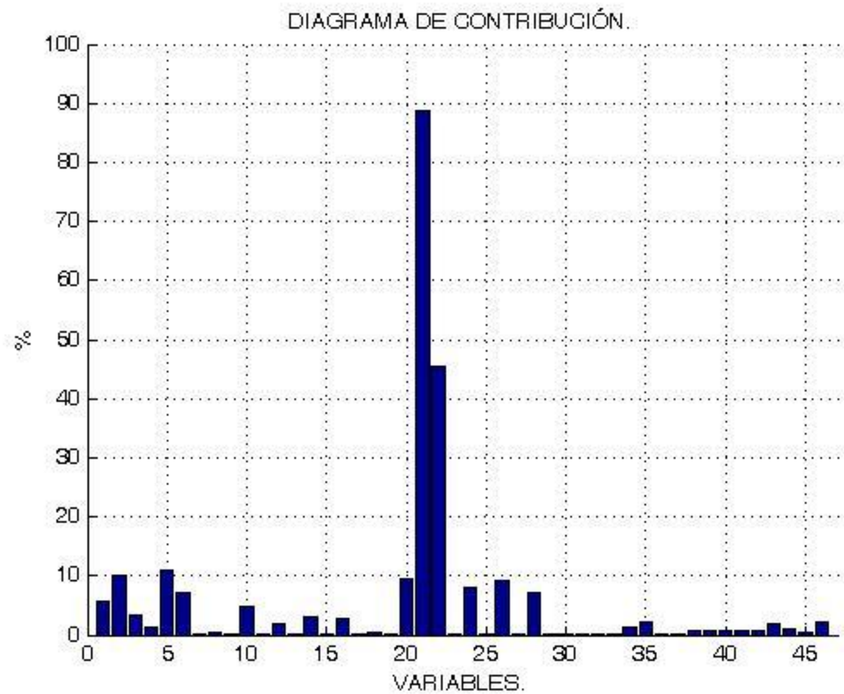


- Variable where the fault is more visible is the variable 12 the level of the first evaporator

Examples. Evaporation Station

- Fault 2.
- Contribution plot

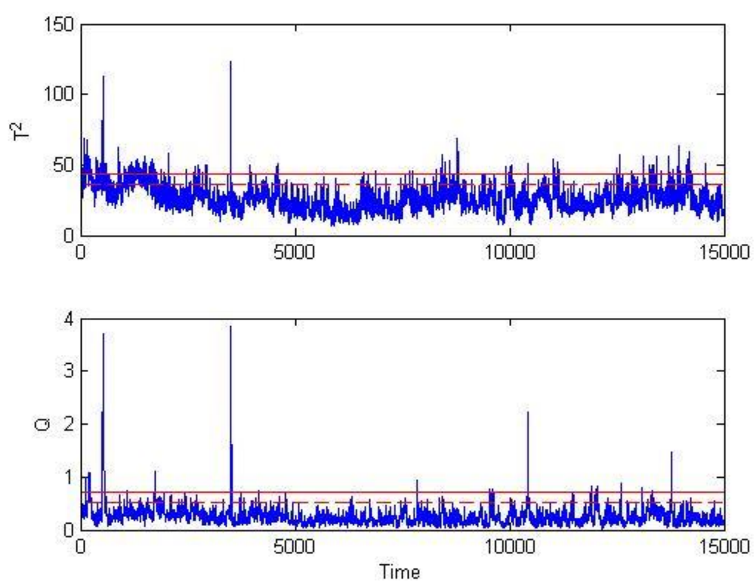


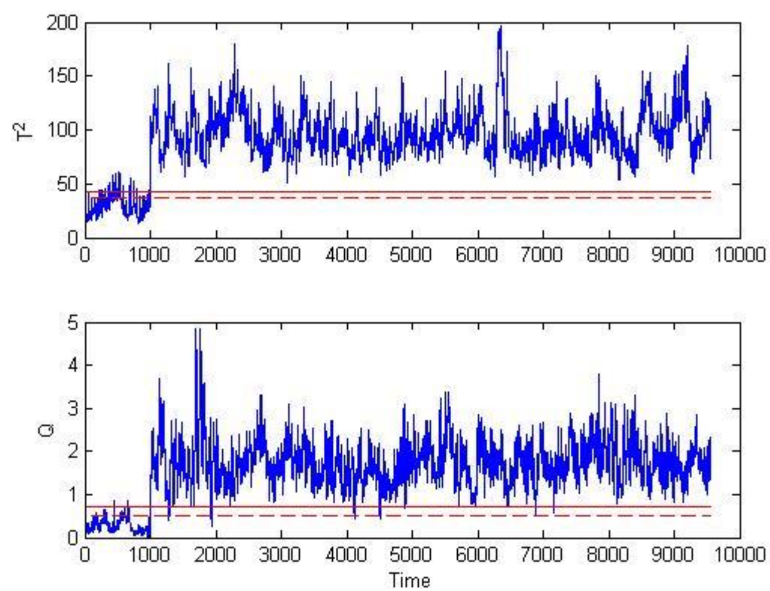


- Variable where the fault is more visible is the variable 21 the level of the third evaporator (IIIb)

Examples. Evaporation Station

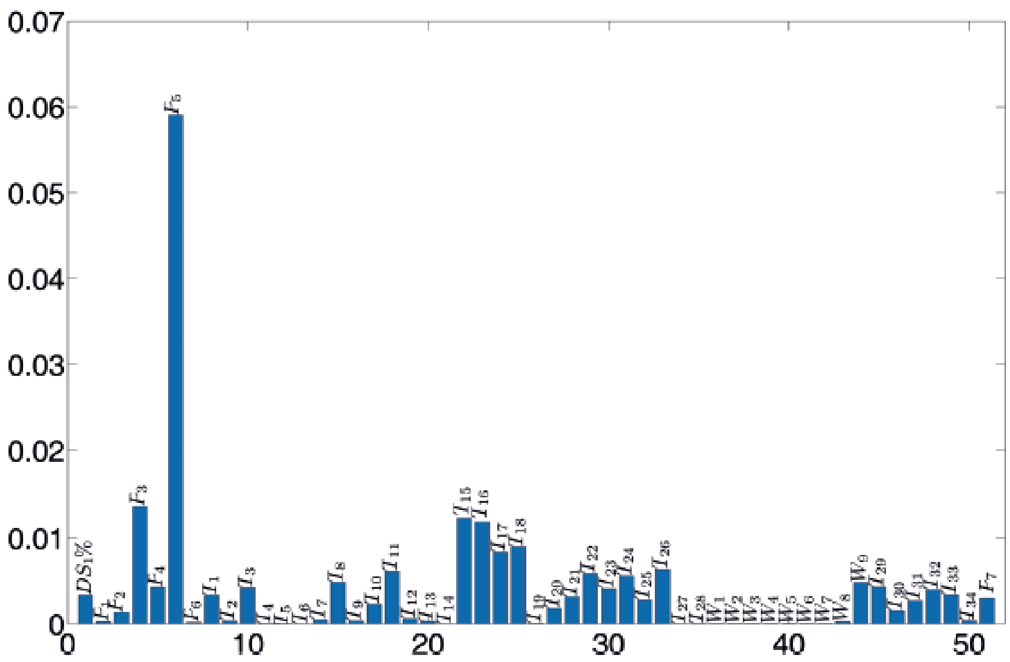
- With real data collected from the plant
  - Only real data from normal operation conditions is collected
  - A fault is simulated adding artificially a constant (5% in magnitude) to the variable 6
  - 52 variables are collected from the plant (temperatures, flows and pressures)



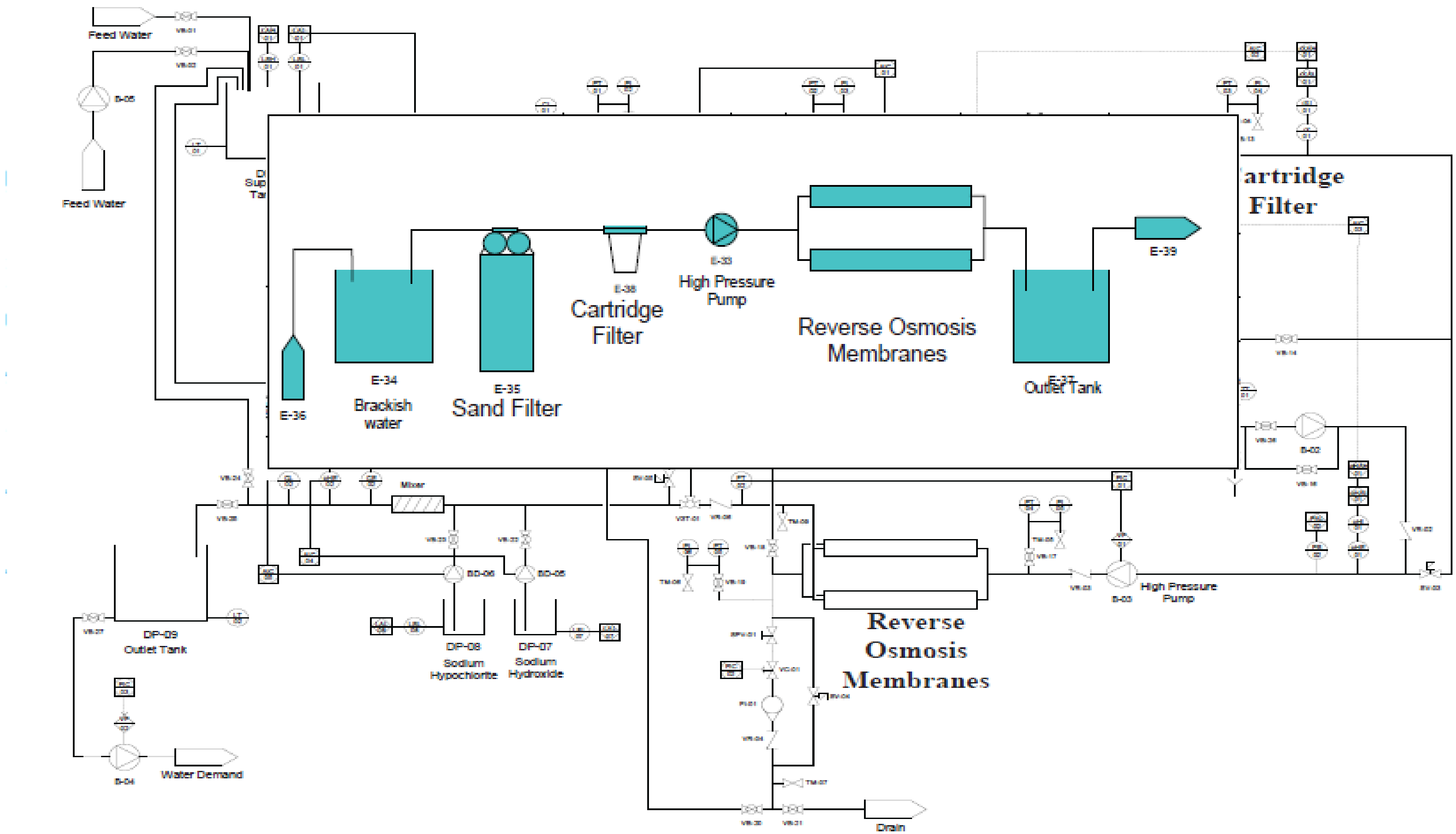


Examples. Evaporation Station

- The contribution plot is

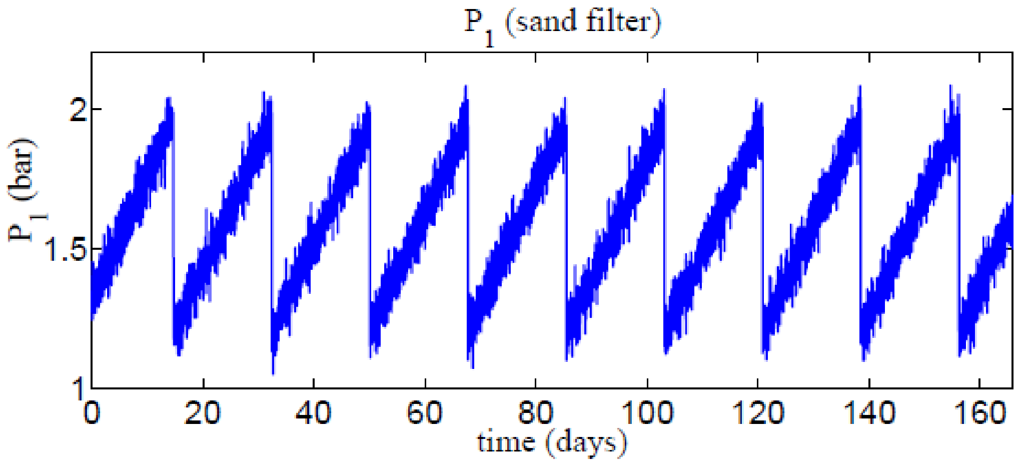


Example: Desalination plant



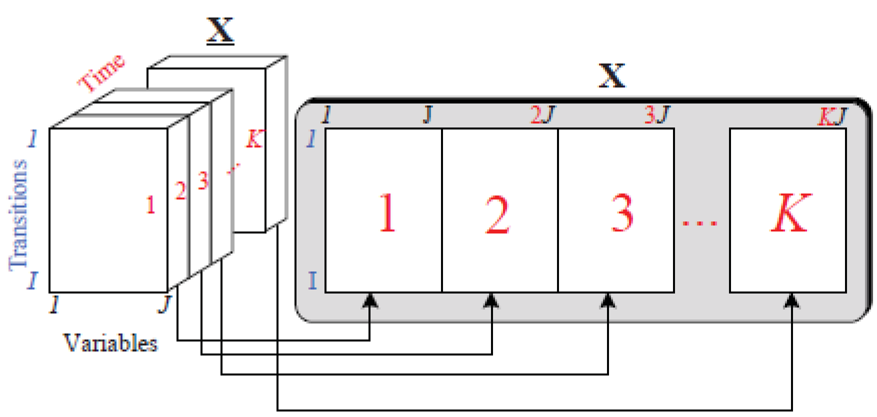
Example: Desalination plant

- The plant is based on reverse osmosis separation process.
- A high pressure is used to force the water through a semi permeate membrane, that retains the salt.
- Two filters are placed before the membrane to eliminate contaminants: the sand and cartridge filters.
- The decrease of performance of membranes and filters is very common due to the several deposits. So cleaning cycles must be run to clean the deposits in order to obtain an optimal plant operation.
- So the process is not strictly in steady state. The variables are as:



Example: Desalination plant

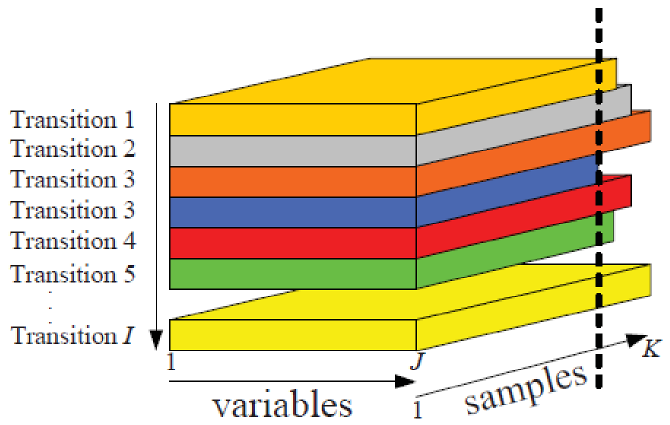
- In this case the time between two cleaning cycles is considered as a batch process.
- A MPCA (Multiway PCA) is used in order to monitor the process.
- Characteristics:
  1. The data collected from the plant have three dimensions X(I x J x K) : i=1,…I batches, j=1,… J, variables, k=1,…, K samples. In order to apply PCA we need a two dimensions matrix=> **unfolding problem** => in this example we use batch-wise unfolding => X(I x JK)



Example: Desalination plant

2. The data collected from each batch can have different number of samples => **data alignment** => different solutions to solve this problem:
   - Indication variable
   - Dynamic time warping (DTW)

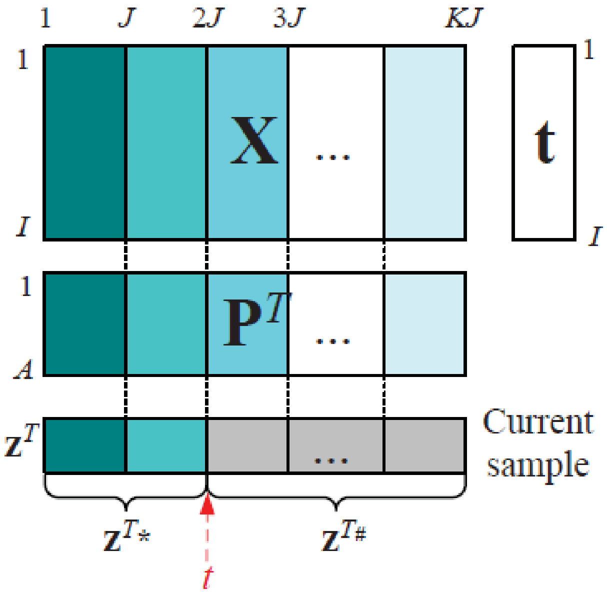
Data alignment



Example: Desalination plant

3. The measured variables between the beginning of the cycle and the current instant t are available, but the measured variables between the current instant t and the end of the cycle are not available. => It is necessary to predict them: **imputation** => some methods to solve this problem.

Imputation



Methods

- Trimmed score method (TRI). $\mathbf{z}^{\#} = 0$
- Known data regression method (KDR): $\mathbf{T}_{1:A} = \mathbf{X}^{*}\mathbf{B}_{1} + \mathbf{U}_{1}$
  - KDR method with PCR.
  - KDR method with pseudoinverse.
  - KDR method with PLS.
- Trimmed score regression method (TSR): $\mathbf{T}_{1:A} = \mathbf{T}^{*}_{1:A}\mathbf{B} + \mathbf{U}$

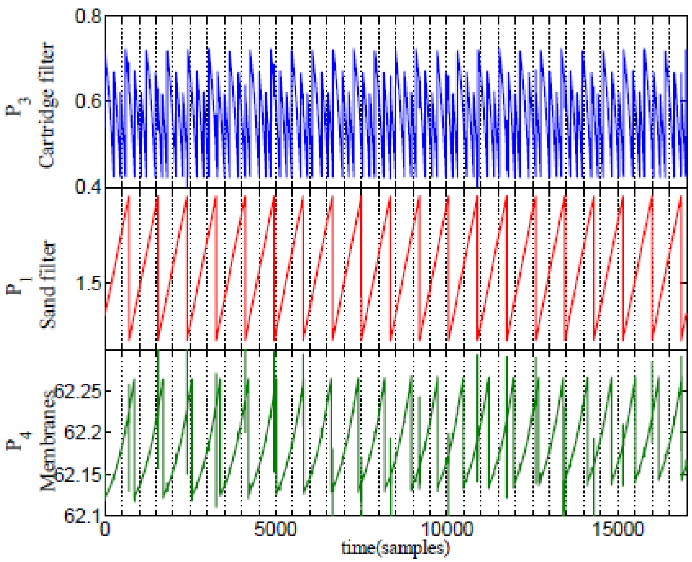
Example: Desalination plant

- Three type of faults where considered:
  - Offset in the pressure sensor in the sand filter input (P_1)
  - Blockage and a breakage in the membrane
- The variables collected form the simulated plant are:

#	Name	Description	Units
1	P_3	Pressure in the cartridge filter input	bar
2	X_{S1}	Total solid concentration in the sand filter input	kg/m^3
3	X_{S2}	Total solid concentration in the sand filter output (cartridge filter input)	kg/m^3
4	P_1	Pressure in the sand filter input	bar
5	P_2	Pressure in the sand filter output (cartridge filter input)	bar
6	X_1	Salt concentration in the sand filter input	kg/m^3
7	P_4	Pressure in the membrane input	bar
8	Q_1	Flow in the sand filter input	m^3/d
9	X_2	Salt concentration in the membrane output	kg/m^3
10	Q_3	Flow in the membrane output	m^3/d
11	Q_2	Flow in the plant output	m^3/d

Example: Desalination plant

Three different PCA models

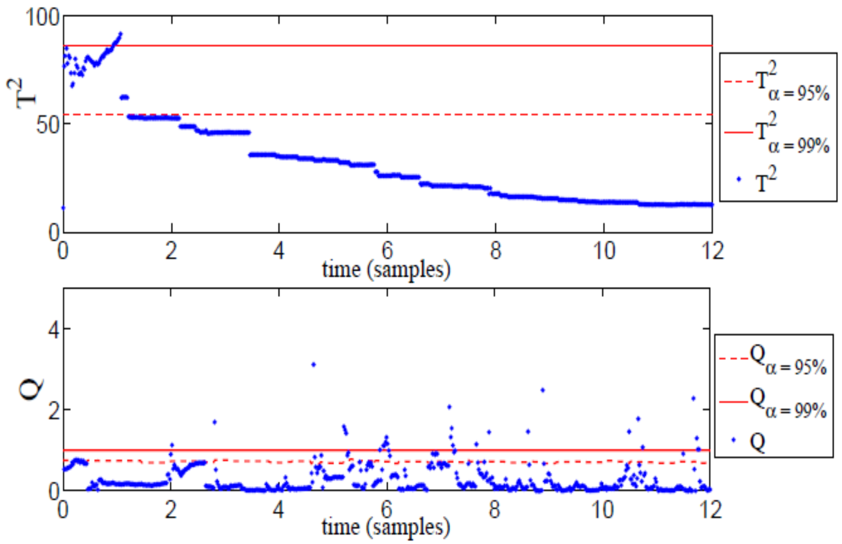


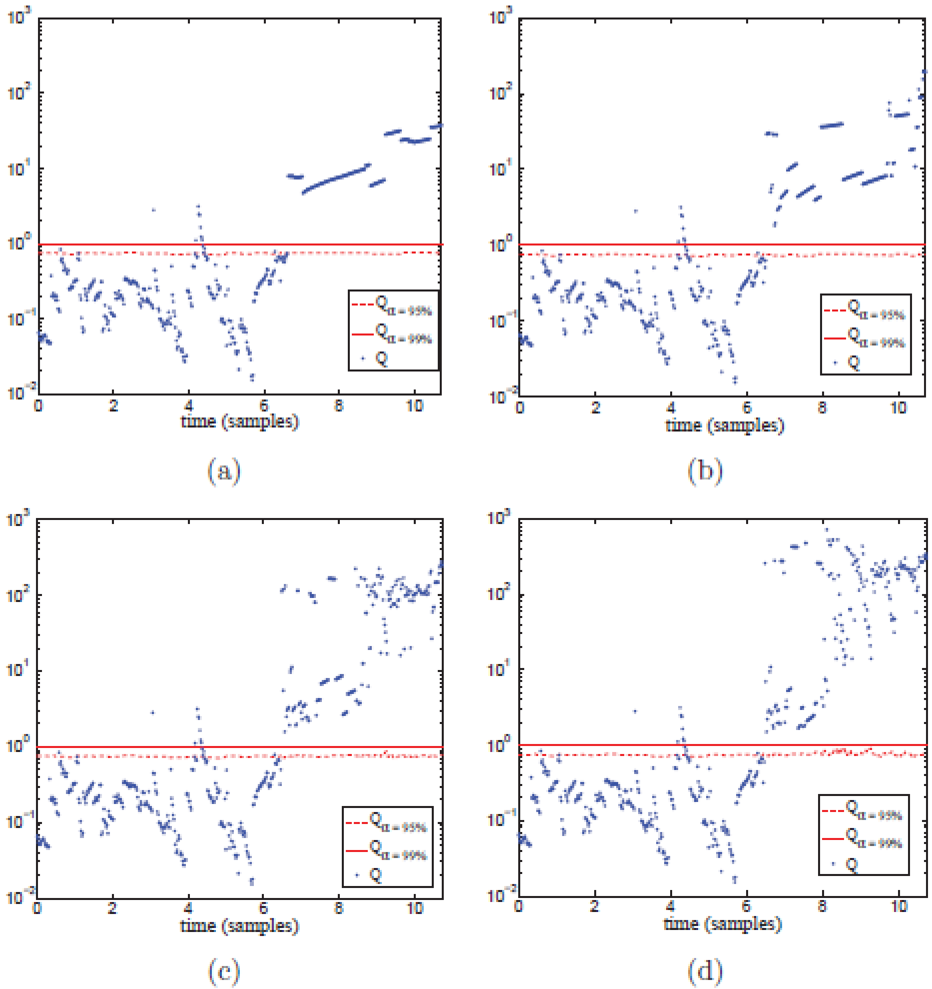
UPCA model	Variables name
Membranes	P_2, X_1, P_4, Q_1, X_2, Q_2, Q_3
Sand filter	X_{S1}, X_{S2}, P_1, P_2, X_1, Q_1
Cartridge filter	P_3, P_2, X_1, Q_1

Example: Desalination plant

- Fulty case

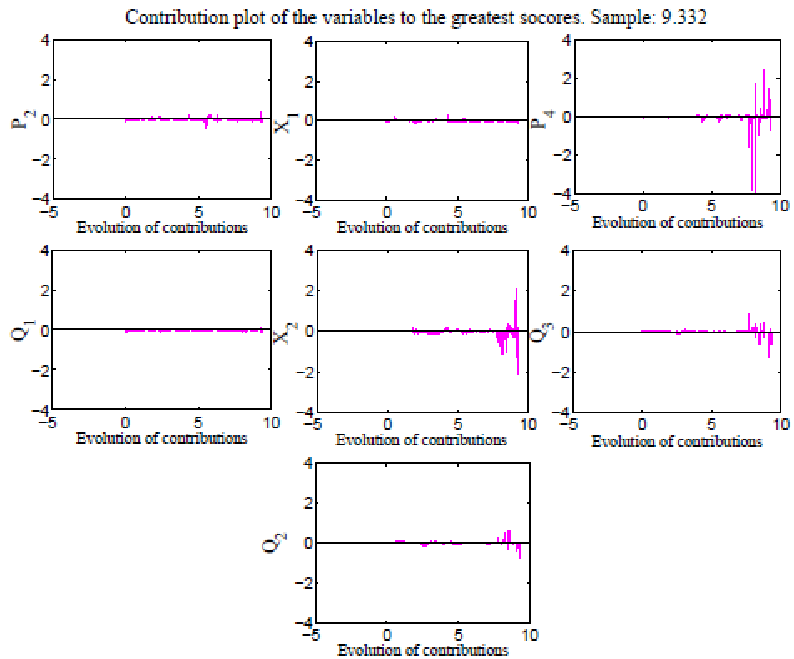
- Nominal case



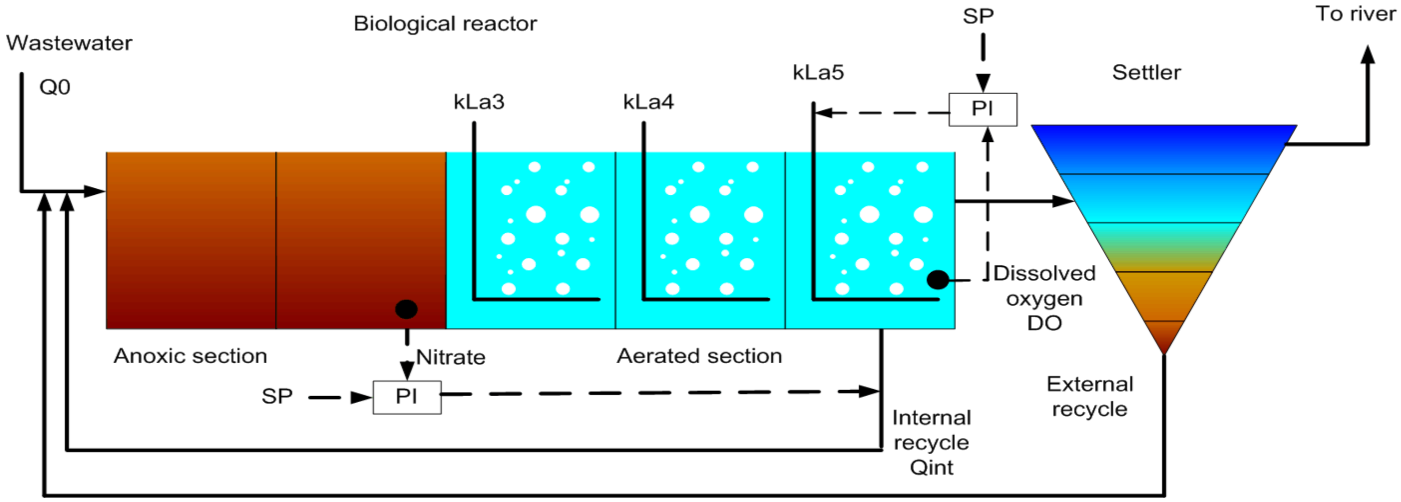


Example: Desalination plant

Variable contributions bar plot to three greatest scores corresponding to a breakage in the membrane.



Wastewater treatment plant: BSM1



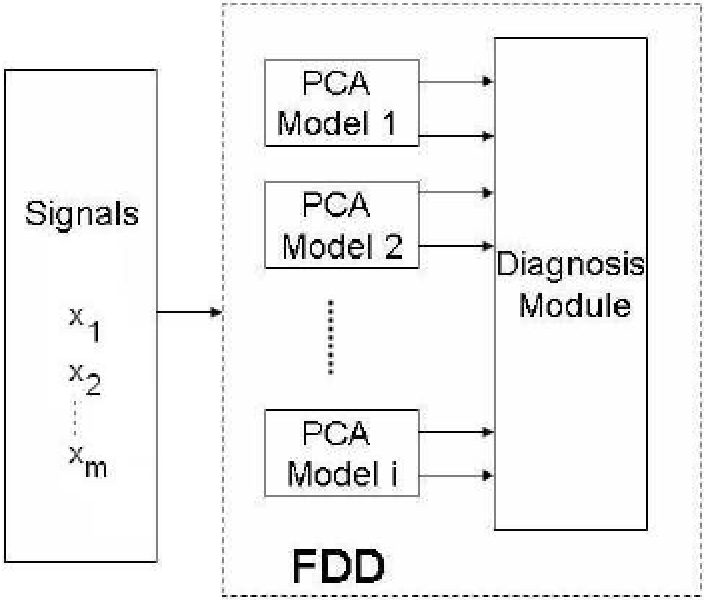
- The benchmark is composed of a two-compartment activated sludge reactor consisting of two anoxic tanks and three aerated tanks.
- And a secondary settler modeled as a 10 layer non-reactive unit
- The objective is to control the dissolved oxygen level in the aerated reactor by manipulation of the oxygen transfer coefficient (KLa5 and to control the nitrate level in the anoxic tank by manipulation of the internal recycle flow rate

Wastewater treatment plant: BSM1

- The system has 13 measured variables.
- Different behaviors can be generated in the plant:
  - **Toxicity shock.** This type of fault can be produced by toxic substances in the water coming from textile industries or pesticides,and causes a reduction in the normal growth of heterotrophic organisms. The fault is simulated as a change in the parameter(μH).
  - **Inhabitation** This fault can be produced by hospital waste that can contain bactericides, or metallurgical waste that can contain cyanide, it causes a reduction in the normal growth of the heterotrophic organisms and an increase in the decay factor of this type of organisms (simulated as changes in the parameters: (μH) and (bH)).
  - **Bulking.** This type of fault is produced by the growth of filamentous microorganisms in the active sludge, i.e., the settling velocity (vsj) is reduced.

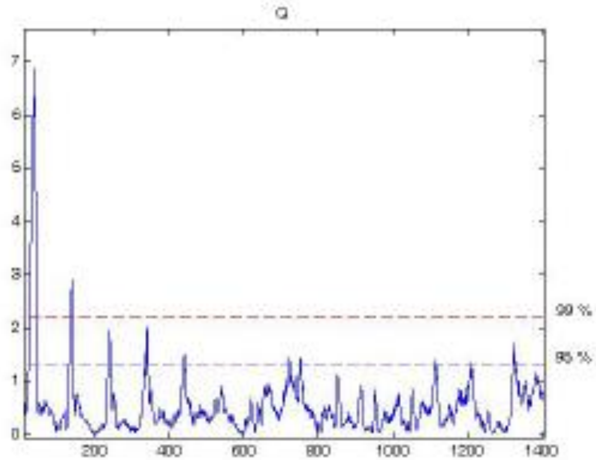
Wastewater treatment plant: BSM1

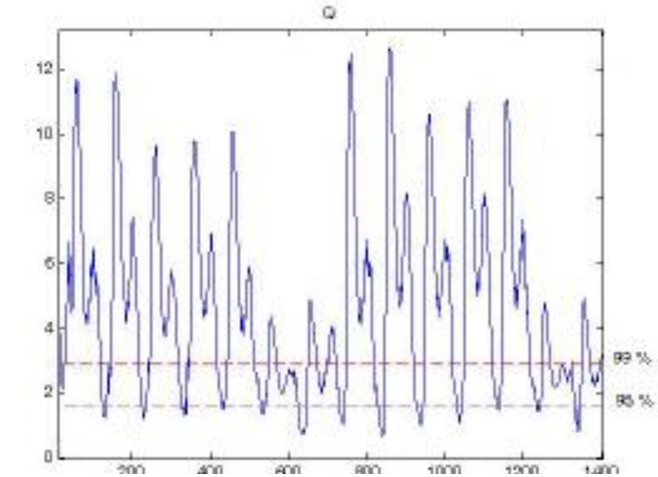
- For fault detection: collect new data from the plant, calculate the statistical T^2 and Q, and compare with its respective threshold
- For fault diagnosis:
  - Contribution plot as before.
  - Specific PCA model for each specific situation (as many models as situations –faults-)

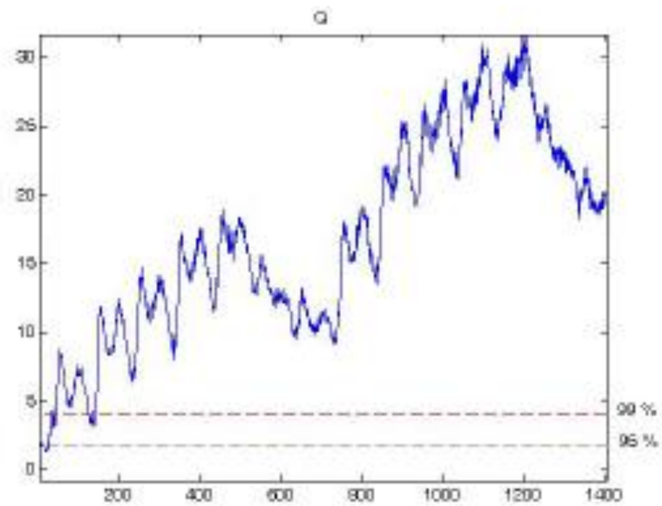


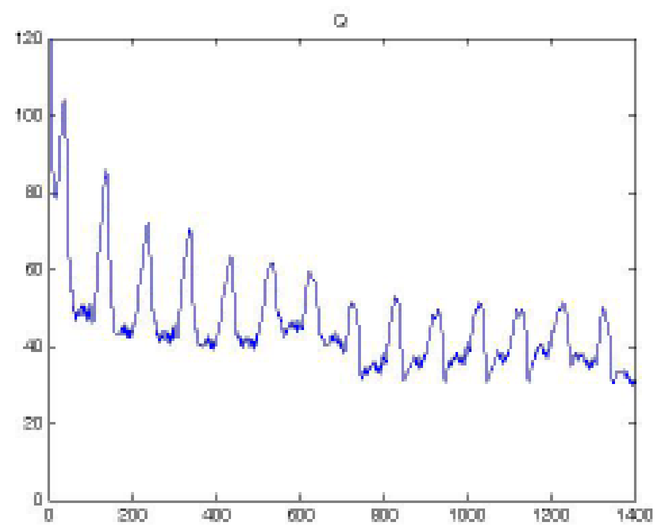
Wastewater treatment plant: BSM1

- Nominal case



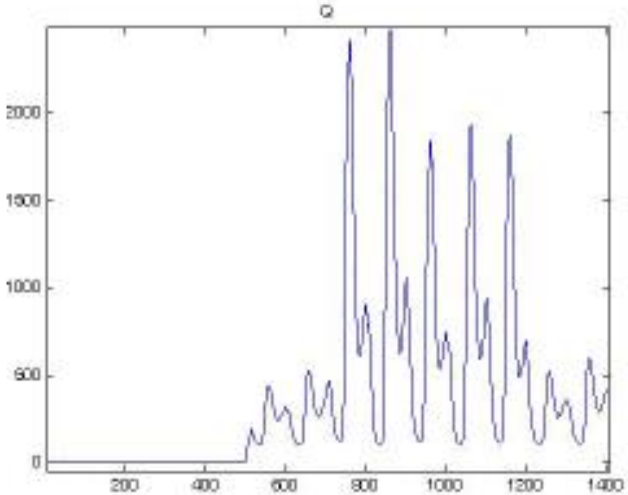


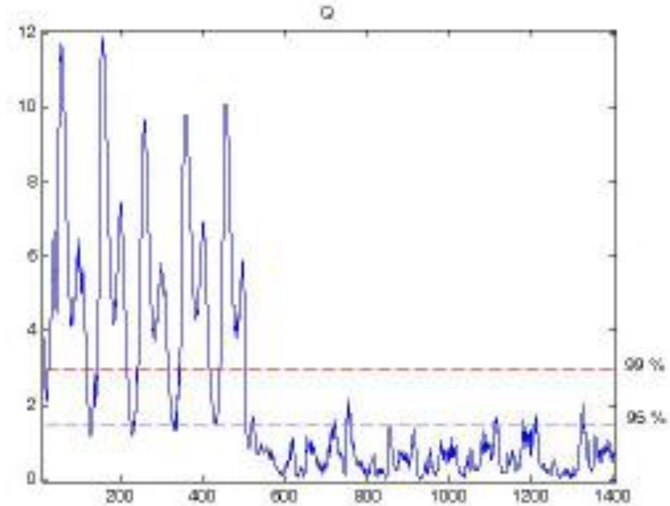


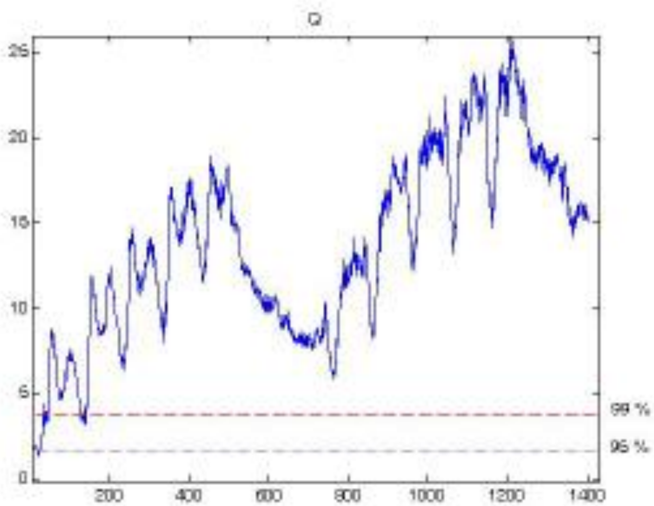


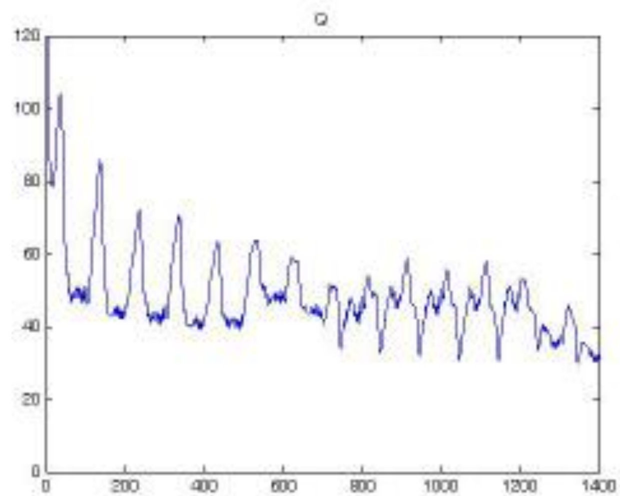
Wastewater treatment plant:BSM1

- Faulty case

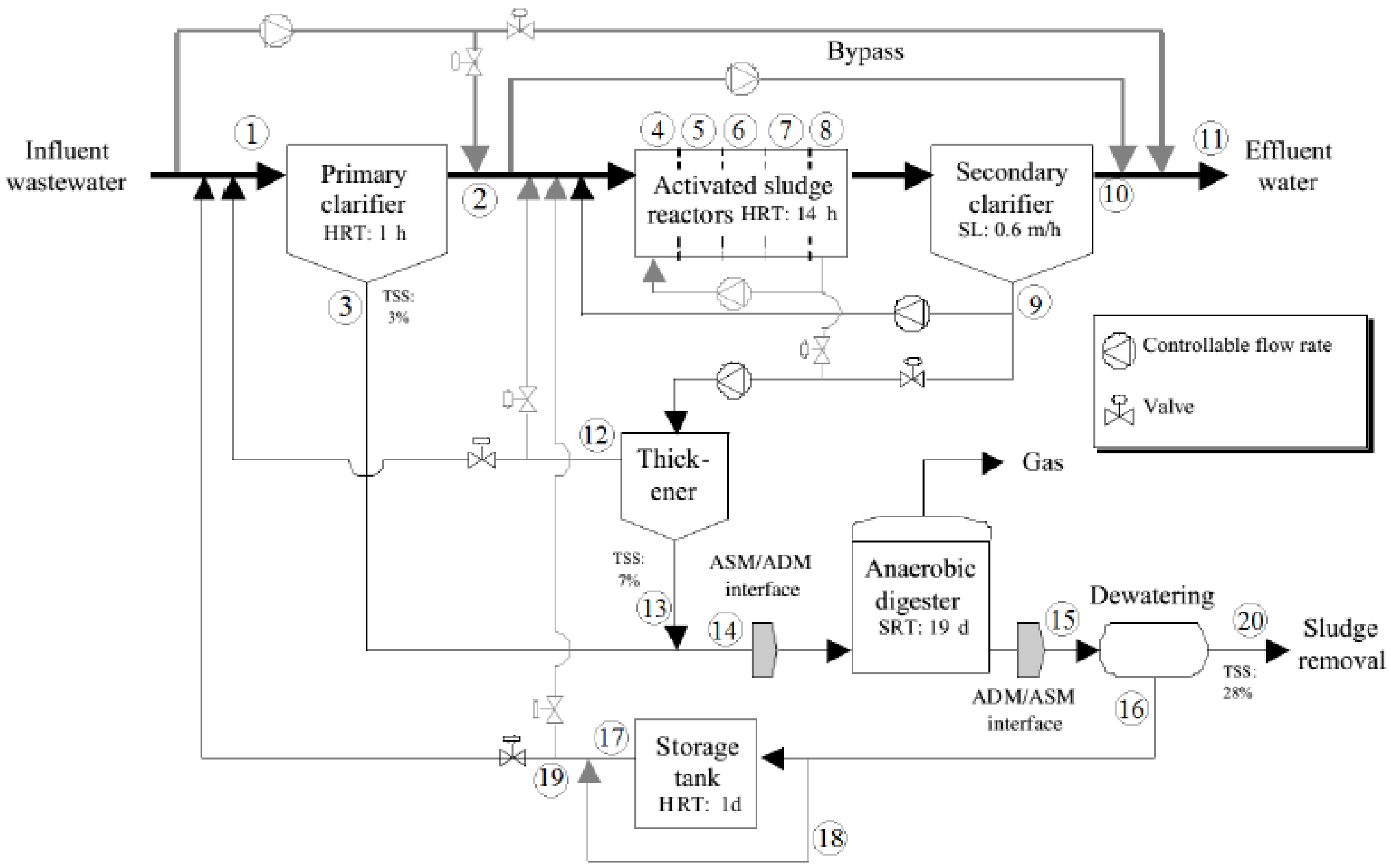








Wastewater treatment plant: BSM2



- More realistic number of variables: 7 variables in each measurement point, there are 20 points => 140 measurements

Wastewater treatment plant: BSM2

Used variables	Variables in the model
COD (Chemical Oxygen demand), g COD.m^{-3}	S_I(inert soluble material), S_S(readily biodegradable substrate), X_I(inert particulate material), X_S (slowly biodegradable substrate)
O_2, g (-COD).m^{-3}	S_O (dissolved oxygen)
Alk (alkalinity)	S_{alk}(alkalinity)
N (nitrogen), g N.m^{-3}	S_{NO} (nitrate and nitrite), S_{NH} (ammonia and ammonium), S_{ND} (soluble organic nitrogen associated with S_S)
SS (solids suspended), g SS.m^3	TSS (total suspended solids)
Flow, $m^3.d^{-1}$	Flow rate
Temp, °C	Temperature

- Several possibilities:
  - Calculate an unique PCA model for all the variables: global PCA
  - To divide the plant into blocks and to calculate a PCA model for each block: DPCA with local models (Distributed PCA)
  - To divide the plant into blocks and to perform some calculations in each block, in order to calculate a global PCA to detect faults in the whole plant => DPCA with QR, CPCA, MPCA, etc.

Wastewater treatment plant: BSM2

- **Faults considered:**
  - F1: A change in the value of the dissolved oxygen measured by a sensor in the aerated reactor of the Activated sludge Reactors unit . This sensor reads a value and sends it to the oxygen control, so if this control works with wrong inputs, it does not introduce the correct amount of oxygen in the reactors.
  - F2: Other failure consists in changing the value of alkalinity in the influent water that enters in the plant. With this it is possible to simulate a change in the influent composition.
  - F3: The other problem was to simulate a malfunctioning in the valves control.
  - The fourth fault (F4), consists in reducing the flow in a pipe to simulate a leak, this was implemented at the exit of the primary clarifier, reducing the flow that arrives to the digester.

Wastewater treatment plant: BSM2

- **Results:** with 16 test, the four faults with different fault magnitude.

Method	Detected faults		Isolated faults		OTI	
	T^2	Q	T^2	Q	T^2	Q
Global PCA	16	16	9	15	1.16	1.26
DPCA (local PCA)	16	16	11	15	0.97	0.86
CPCA	16	15	0	0	3.67	0.84
Merge PCA	16	16	10	13	3.82	5.02
DPCA (QR)	13	16	6	12	4[illegible]	[illegible]
DPCA clustering	16		11		[illegible]	

Method	Detection time		False alarms	
	T^2	Q	T^2	Q
Global PCA	1429.8	404.6	0 %	6.25%
DPCA (local PCA)	588.6	5.13	6.25%	6.25%
CPCA	2083.8	1151.8	0	0
Merge PCA	9.6	448.8	62.5%	100%
DPCA (QR)	108.92	166.8	25%	100%
DPCA clustering	21.83		0%	

Wastewater treatment plant: BSM2

- **Results:** with a fault in the oxygen sensor in the fourth aeration thank, i.e., in the block seven.

  - DPCA with local PCAs

Node	T^2			Q		
	Most contributing variable	Contribution	Detection time (sample)	Most contributing variable	Contribution	Detection time (sample)
1						
2	O_2	3.6992	19372	O_2	4.4213	18817
3	O_2	4.7949	19372	O_2	5.5801	18816
4	O_2	1.4751	19041	O_2	6.5392	18802
5				N	2.1836	18840
6	O_2	6.2449	18802	O_2	7.4112	18802
7	O_2	**47.0129**	**18802**	O_2	**24.2129**	**18802**
8	O_2	3.9788	18802	O_2	5.1652	18802
9	COD	1.2076	18810	O_2	3.6964	18812
10	O_2	3.0684	18820	O_2	3.7909	18817
11	O_2	2.8105	18819	O_2	3.9896	18817
12	O_2	0.9815	18811	O_2	3.6964	18812
13	O_2	3.2393	18813	O_2	3.1254	18811
14				O_2	3.9586	18813
15	COD	0.9238	26893			
16	Flow	3.8055	26855			
17						
18	Flow	3.8055	26855			
19	Flow	3.8055	26855			
20	S_{alk}	1.4748	26943			

  - DPCA with QR

Node	T^2			Q		
	Most contributing variable	Contribution	Detection time (sample)	Most contributing variable	Contribution	Detection time (sample)
1						
2						
3				O_2	0.5877	19433
4				O_2	1.0174	18802
5				O_2	0.6005	18803
6	Temp.	1.47e-05	18802	O_2	1.4518	18802
7	O_2	**3.29e-05**	**18802**	O_2	**3.2783**	**18802**
8	Temp.	1.43e-05	18802	O_2	1.3223	18802
9	Temp.	1.18e-05	18807	O_2	0.5496	18809
10	Temp.	9.05e-06	18809	O_2	0.5149	18810
11	Temp.	1.24e-05	18812	O_2	0.5149	18810
12	Temp.	1.21e-05	18808	Flow	0.6435	18808
13	Temp.	9.65e-06	18815	O_2	0.6243	18810
14				O_2	0.5408	18812
15	COD	1.23e-05	27022			
16						
17						
18						
19						
20	S_{alk}	1.85e-5	36192			

Wastewater treatment plant: BSM2

- **Results:** with a fault in the oxygen sensor in the fourth aeration thank, i.e., in the block seven.

- DPCA with clustering

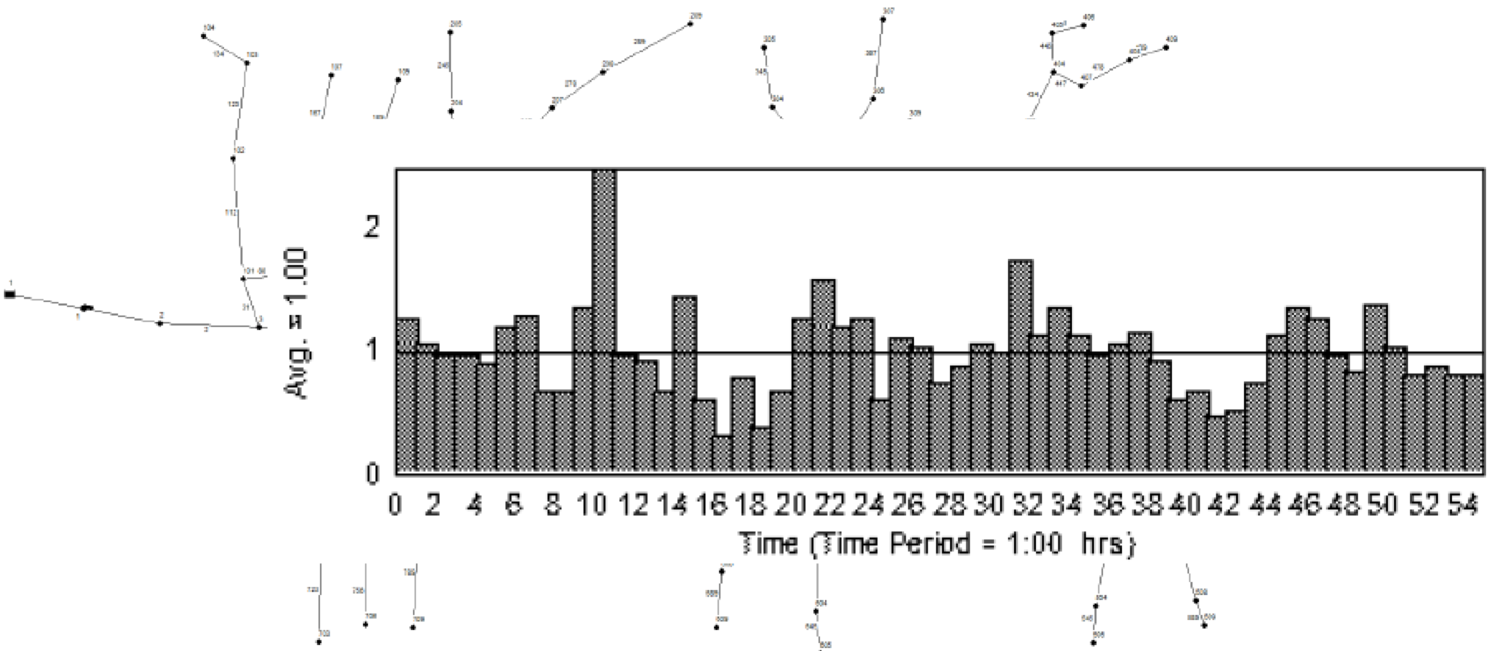
# of block	Clusters over the limit	Observations over the limit	Most contributing variable	Contribution
4	8	2	O_2	3.4384
6	7	2	O_2	3.7306
7	**6**	**2**	O_2	**5.7034**
8	3	2	O_2	2.5782

- Global PCA

T^2			Q		
Most contributing variable	Contribution	Detection time (sample)	Most contributing variable	Contribution	Detection time (sample)
$O_2, Settler-underflow$	0.4538	19430	O_2, $Reactor4$	24.2129	18802

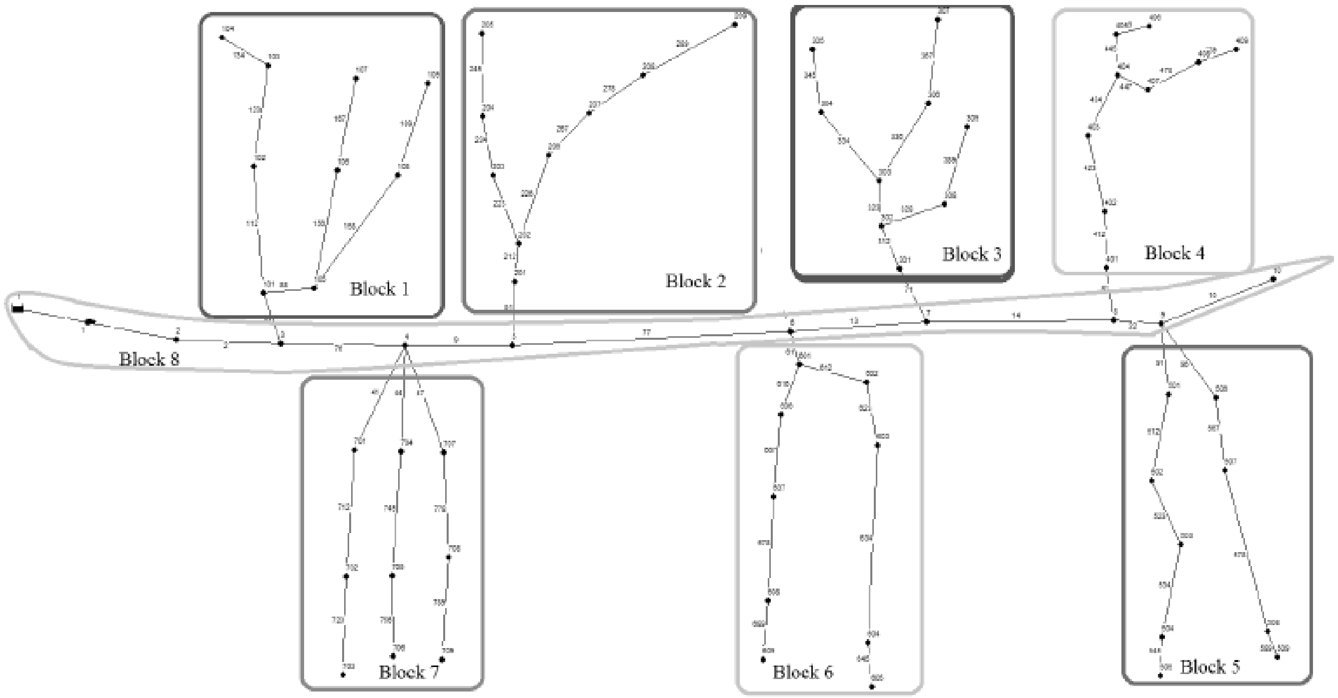
Water distribution net

- The water distribution net was modelled using EPANET software
- Includes a pump that takes the water from reservoir, and a central pipe with branches that distribute the water to the points of consumption
- The water demand is not constant



Water distribution net

- In each node of consumption there are four variables to measure and in the pipes it is possible to measure 5 variables.
- There are 72 points of consumption and 72 pipes, resulting in 648 variables => divide the networks in 8 blocks
- 3 faults: fault in the bomb, in a pipe and in the injection of a contaminant in a node



Water distribution net

- **Results:** with 9 test, the three faults with different fault magnitude.

Method	Detected faults		Isolated faults		OTI	
	T^2	Q	T^2	Q	T^2	Q
Global PCA	0	6	0	2	1.31	5.11
DPCA (local PCA)	5	6	3	6	1.62	4
CPCA	1	1	0	0	2.67	0.44
Merge PCA	2	6	2	4	5	2.1
DPCA (QR)	4	4	3	3	5	1.87
DPCA clustering	5		5		0	

Method	Detection time		False alarms	
	T^2	Q	T^2	Q
Global PCA	-	1.5	0 %	0%
DPCA (local PCA)	5.67	1.33	0%	0%
CPCA	34	10	0	0
Merge PCA	10.5	5.67	0%	0%
DPCA (QR)	6.75	4.5	0%	0%
DPCA clustering	5.4		0%	

Water distribution net

The results obtained when fault F1 (appearance of a contaminant in one point of the net) occurs at time 150h consisting of $2.66l/s$ of water with $1mg/l$ entering through node 6.

CENTRALIZED PCA

	T^2			Q		
	Most contributing variable	Contribution	Detection time	Most contributing variable	Contribution	Detection time
				Node 6 quality	34.7795	152

CPCA

	T^2			Q		
	Most contributing variable	Contribution	Detection time	Most contributing variable	Contribution	Detection time
				$Node408$ $quality$	1.4991	160

DPCA

Node	T^2			Q		
	Most contributing variable	Contribution	Detection time	Most contributing variable	Contribution	Detection
1						
2						
3	$Link71$ $quality$	2.4856	154	$Link71$ $quality$	17.8988	153
4	$Link412$	4.6028	157			
5				$Link91$ $quality$	31.0952	161
6				$Link61$ $quality$	5.8054	155
7						
8	**Node 6 quality**	10.9376	152	**Node 6 quality**	34.7795	152

MERGED PCA

Node	T^2			Q		
	Most contributing variable	Contribution	Detection time	Most contributing variable	Contribution	Detection time
1						
2						
3	$Link71$ $quality$	9.4641	154	$Link71$ $quality$	3.4109	153
4	$Link81$ $quality$	16.2757	156	$Link81$ $quality$	3.3284	155
5				$Link91$ $quality$	3.0430	161
6				$Link61$ $quality$	2.2792	165
7						
8	**Node 6 quality**	16.5282	152	**Link 6 quality**	7.5844	152

PCA Extensions

- There are many different variations to the classical PCA-based fault detection method.
- The different proposed methods present different improvements and considerations in order to reduce the number of false alarms, to detect consecutive faults or to detect faults in transient states.
  - Dynamic PCA (DPCA)
  - Adaptive PCA (APCA)
  - Recursive PCA (RPCA)
  - Multiscale PCA (MSPCA)
  - Exponentially weighted PCA (EWPCA)
  - PCA using external analysis (PCAEA)
  - Non-linear PCA (NLPCA) with neural networks or with kernels: KPCA
  - Robust PCA , etc

Other MSPC methods

- Pattern recognition-based methods. Fisher discriminant analysis (FDA).
- Partial least squared (PLS).
- Independent component analysis (ICA).
- Correspondence analysis (CA).
- Canonical variate analysis (CVA).
- Etc.

Hybrid methods for FDI

- Any FDI method is the best for every application
- In each situation it is necessary to choose the most adequate FDI method: based on models or based on data.
- Also a best solution is the combination of methods, i.e., to implement an hybrid method.
    - Using models to generate the residuals and PCA to evaluate them.
    - Use neural networks to calculate the non-linear model and the residuals and to evaluate them with PCA
    - Use models to calculate the residual and neural networks to evaluate them.
    - Etc.

Basic Bibliography

- J. Gertler (1998), Fault detection and diagnosis in Engineering Systems, Marcel Dekker, New York
- J. Chen and R.J. Patton (1999), Robust model-based fault diagnosis for dynamic systems, Kluwer Academic Publishers
- E. L. Rusell, L.H. Chiang, R.D. Braatz, Data driven techniques for fault detection and diagnosis in chemical processes, Springer-Verlag col. Advances in Industrial Control, 2000
- M. Blanke, M. Kinnaert, J. Lunze and M. Staroswiecki (2003). *Diagnosis and Fault-Tolerant Control*. Springer
- J. Korbicz, J. M. Koscielny, Z. Kowalczuk and W. Cholewa (2004). *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer
- R. Isermann (2006). *Fault Diagnosis Systems*. Springer
- Model based Fault Diagnosis Techniques, S.X. Ding, Springer, 2008
- Etc.