

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

7,000

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



AI Overview: Methods and Structures

*Erik Dahlquist, Moksadur Rahman, Jan Skvaril
and Konstantinos Kyprianidis*

Abstract

This paper presents an overview of different methods used in what is normally called AI-methods today. The methods have been there for many years, but now have built a platform of methods complementing each other and forming a cluster of tools to be used to build “learning systems”. Physical and statistical models are used together and complemented with data cleaning and sorting. Models are then used for many different applications like output prediction, soft sensors, fault detection, diagnostics, decision support, classifications, process optimization, model predictive control, maintenance on demand and production planning. In this chapter we try to give an overview of a number of methods, and how they can be utilized in process industry applications.

Keywords: process industry, artificial intelligence (AI), learning system, soft sensors, machine learning

1. Introduction

During the 80th AI was a hot topic both in the academia and industries. Many researchers were working a lot with development of methods for diagnostics, simulation and adaptation of models. Artificial Neural Networks (ANN) were being implemented in real applications such as e.g. soft sensors to predict NO_x concentration in exhaust gas from power plants. Still there was quite some “over-selling” and the enthusiasm for AI in the future was assumed to be useful tomorrow. But it took much longer to get the systems robust enough to be used and fast enough to be applicable in on-line applications. After year 2000, systems started to reach a more mature state and we got IBMs Watson, that could beat the Jeopardy master. Later the Google tool could beat the “Go-master”, a very complex Chinese game. This has changed the perception of AI. It is still similar type of tools as were developed during the 80th, but now they were refined a lot and hardwires has been developed dramatically. This has given us a much more positive perception of what can be done, and a lot is now being implemented. Still there is a risk for over-selling, as the tools are normally not that “intelligent” as we normally think of when we talk about Intelligence. But we are closing the gap day by day.

Concerning use of AI in process industry, we cannot just take the tools and hope they will fix everything. It is still important to identify “what is the problem to solve”? With Jeopardy the goal is to be good at Jeopardy, but what is the goal in process industry? It should be to increase production, reduce process variations,

implement maintenance on-demand and give operator support. It also means to coordinate and optimize production lines as well as complete plants and later on complete corporations. It also means to adapt to changing customer demands, support in development of new products with production lines as well as handle new business models. These different functions demand quite different tools and thus we will not use only one but several. Often Machine learning is considered being “the tool”, but often there is not data available to implement ML, especially not when starting a new production line. To implement new tools, it is also very important to pre-treat data. You have to sort data in “normal variations” or “anomalies”. You may need to filter data with moving windows, but in different time perspectives. We need to do data reconciliation to handle drifting sensors. And you need to integrate all levels from orders to production planning down to coordinated and optimized production. In this chapter we will discuss a number of different methods as well as discuss integration between the different levels. Over the years many researchers have investigated different AI techniques for different process industrial application. A comprehensive review on different AI models applied in energy systems can be found in [1]. Applications of different AI tools based on simulation models in pulp and paper industry has been presented by researchers including Dahlquist [2–5]. Applications in power plants have been presented in many articles including Karlsson et al. [6–8]. In Karlsson et al. [9] a general discussion is made on how to make better use of data including pretreatment of data. Adaptation to degeneration in process models by time is discussed in Karlsson et al. [7]. [10] conducted an extensive review on different AI based soft sensors in process industries.

1.1 Similarities between AI and how the brain works

The mathematicians developing especially ANN have been looking a lot on how the brain works. In **Figure 1** we see a principal picture of a human.

Running in a forest: The brain stores many different factors locally by “tuning many soft sensors”. During the night strength of connections are enhanced for the most important functions, while other less important connections are eliminated. Some information is used for direct control. Others is stored for use later on.

If it is rainy when you run there is a general feeling that “this was not so nice”. Everything else happening in the forest then will be “colored” by this in your memory, aside of concrete thing like if you meet someone, like a friend, during the run.

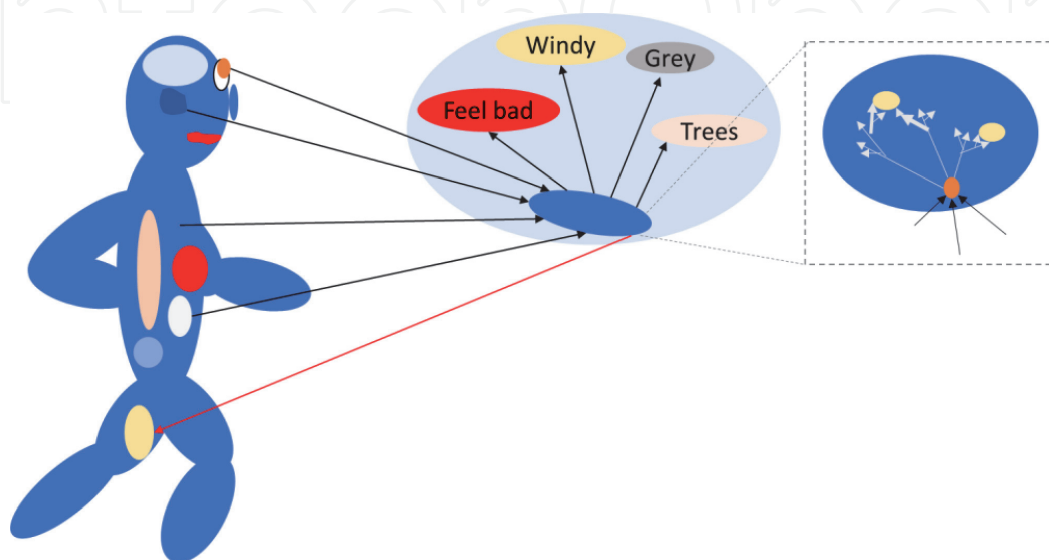


Figure 1.
How a human handle input from the surrounding.

Short term memory: Dorsolateral prefrontal cortex controls information stream from sensors. Skull lobe is for attention. Ventrolateral prefrontal cortex sort information into useful or not useful info. Supplementary motor area (SMA) repeat new memories all over.

Long term memory: Hippocampus and nearby areas in medial temple globe are essential for long term memory. Facts are stored. Small brain and basial ganglia contain procedural memory, like how to bike or swim.

A human may have approximately 120 billion nerve cells. Each connect to hundreds of other cells. Some connections enhance while other decrease signals. Very complex interactions where connections are established and broken continuously. No exact values or memories exist for control, but diffuse input give diffuse output, but with different feed-back mechanisms. The Swedish Nobel Prize winner Arvid Carlsson [11] found out the mechanism of how signals are transferred from the dendrite of one cell to the axon of the next, where complex feed-back mechanisms enhance a connection and thereby also enforced a memory by changing the easiness of transferring new signals. He explored how dopamine works as a signal substance, which we now know is of highest importance in the brain. By back-propagation in ANN we try to simulate this mechanism (**Figure 2**).

Input to the brain is sorted in Amygdala and hippocampus. Signals are sent to different part of the brain Here different signals are enhanced or decreased depending on previous experiences in many different “soft sensors”, built up with tuning of Ca-channels working as parameters in a polynom. “= enhancement factors”. The situation is triggering memory build up. All control is “diffuse” using many different “diffuse” measurements. Different individuals have different sensitivity and number of different sensors like sense for bitterness, sugar, pain etc. Soft sensors get input and react with output to other soft sensors. Signals are sent to direct different biochemical processes like when fear - increase production of Adrenalin and Cortisone. This in turn is affecting many other hormones and proteins etc. Also, microbiome in the stomach and skin send input to the brain on how these organs perform. When you run, the body feel good and e.g. endorphins are produced enhancing performance of stomach, muscles etc. Serotonin levels,

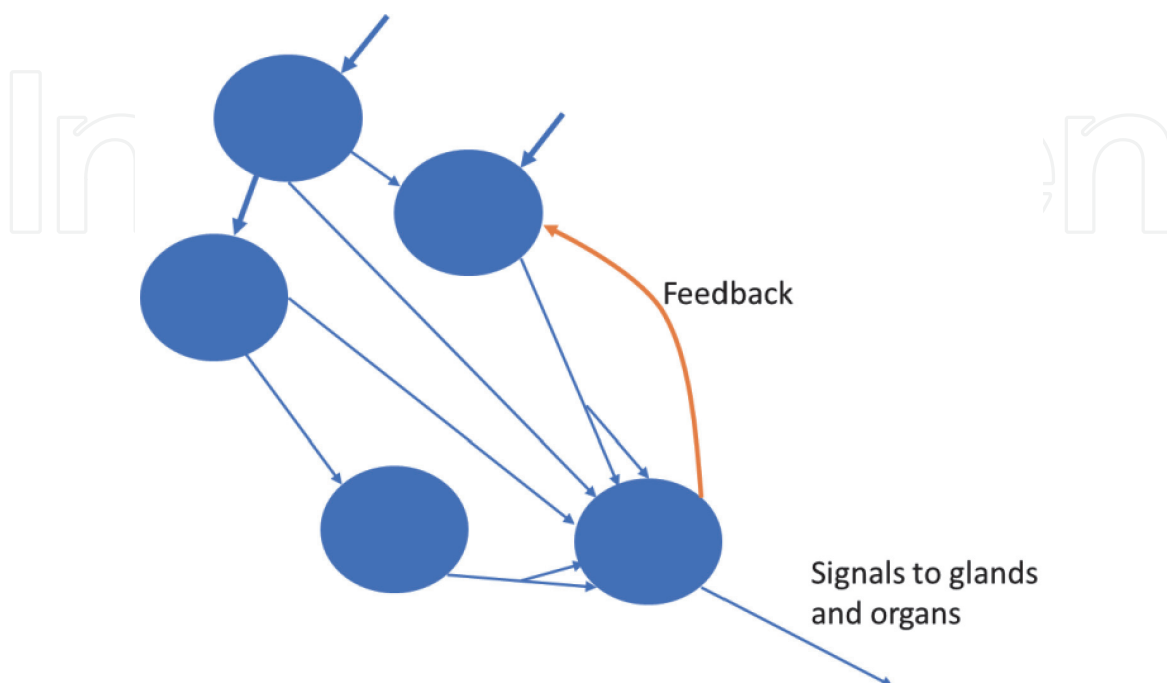


Figure 2.
Signals flow in the brain – Many connections and feed-back enhance learning.

gibberellins, insulin, cortisone etc. are interacting and tuning each other, but with influence “from the side” by other sensor inputs. The brain is interacting with all this. This is also the basic concept to mimic in “deep learning”.

If we try to transfer this picture into a control system, it can look like below in **Figure 3**.

We start with sorting out “outliers” in pre-processing. This is what the brain does with information from the eye etc. The outliers can be used for anomaly detection. This is principally what is done in Amygdala. We then compare predictions from simulators and soft- sensors to measurements. We trend differences developed by time. Refined data are used for model building and adaptation of models. The models are used for soft sensors, diagnostics, control etc. We also make conclusions in decision a tree from previous experience and identify optimal action to take in different time perspectives. In the brain this is done by utilizing previous experience in a way where we try to “make sense”. This means that we replace missing data with what is reasonable. In our computer system we do this by data-reconciliation using e.g. solving an equation system of physical models to get a best fit. We then take actions by control of many different functions more. In the body, this means e.g. control of sugar content in the blood, release of adrenalin to meet threats or melatonin to make you tired and go to sleep. We learn buy tuning soft sensors and decision trees with the new information just as the brain does, but where the brain is very much more complex than what we can handle today.

1.2 Market aspects

IndTech’s market, i.e. Products and systems for industrial digitization and automation in the world are worth around USD 340 billion in 2016/2017 and have an average growth rate of 7–8 percent. The area can be divided into two parts: IT (industrial IT) and OT (operational technology). The share that can be categorized into industrial IT is about USD 110–120 billion. The remaining USD 220 billion is operational technology for the factory floors and in the field. It, in turn, is traditionally divided into discrete automation (about 45 percent) and process automation (about 55%). OT includes various types of industrial control systems (ICS) and field equipment such as instrumentation, analysis, drive systems, motors, robots and similar.

For the future of AI, we can see that this comes deeply into all these industrial market segments, but also far beyond as not only for industrial applications.

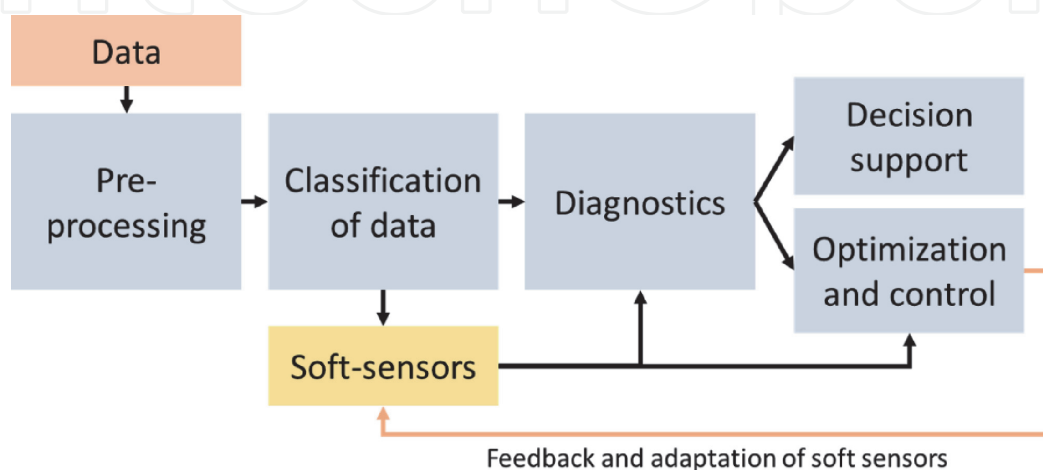


Figure 3.
Principal diagram of signal processing in a “learning system”.

The tools thus will be developed for one application, but then will be used also for other applications most probable.

2. Different AI methods

There are many different methods developed. Some of them are very similar or aim to solve the same type of problems. If we look at Machine learning (ML), we have e.g. Regression. Artificial Neural Networks (ANN), Support Vector Machines (SVM), Principal Component Analysis (PCA), Partial Least Square regression (PLS) and etc. They both aim to sort different variables into group that correlate to different properties or faults.

PLS and ANN, both are very useful to create soft sensors. Deep learning is a sophisticated version of the ANN, but with the goal to produce models that can do much more than just be a soft sensor, which predicts one or more qualities. Examples of soft sensors is to predict strength properties of paper from e.g. NIR data and process variable values in paper machines, amount of different kind of plastics in Waste combustion plants or protein content in cereals in agriculture from NIR spectra. The deep learning on the other hand can be used to teach a robot to pick out machine components that are scrapped from a conveyor belt for instance. This then includes image pattern analysis from camera monitoring of the parts passing.

A selection of different tools is listed in **Table 1**.

2.1 Machine learning methods

Machine learning methods principally use a lot of process data measured preferably on-line, and identify correlation models from the data, which can be used for different purposes like soft sensors, anomaly detection and others.

There are several different machine learning methods. Some are correlating a specific property to process data. Reinforcement learning is described in e.g. Gattami Ather [12]. It is used in problems where actions (decisions) have to be made and each action (decision) affects future states of the system. Success is measured by a scalar reward signal and proceed to maximize reward (or minimize cost) where no system model is available. One example of this technique is deep reinforcement learning which was used in AlphaGo that defeated the World Champion in Go. Here a Q function is approximated with a deep neural network. Minimizing the loss function with respect to the neural network weights w is made as given below

$$l = (r(s, a) + \delta \sup Q(\acute{s}, \acute{a}, w_-) - Q(s, a, w))^2 \quad (1)$$

-
- Gaussian Process Regression (GPR)
 - Partial Least Square (PLS) Regression
 - Principal Component Analysis (PCA)
 - Artificial Neural Networks (ANN)
 - Support Vector Machines (SVM)
 - Gray box models
 - Physical models, MPC – model predictive control
 - Bayesian networks (BN)
 - Gaussian Mixture Model (GMM)
 - Reinforcement Learning
 - Google algorithm – search engines
-

Table 1.
A selection of different common AI-tools.

If the system is deterministic the model is given by

$$S_{k+1} = f_k(s_k, a_k) \tag{2}$$

If the system is stochastic the model is given by

$$P(s_{k+1}|s_k, a_k) \tag{3}$$

$f_k(s_k, a_k)$ is a scalar valued reward.

In Werbos Paul: A Menu of Design for reinforcement learning over time [13] reinforcement methods are described more generally.

2.2 Soft sensors

It is interesting to create soft sensors by creating models correlating process measurements on-line to quality measurements from samples analyzed at lab. The soft sensor then can be used to predict the quality property on-line from feeding the on-line measurements into the soft sensor model. There are several different methods for the regression, and a number of alternatives are given in **Figure 4** below.

In **Figure 5** we see how the data flow can look like for data collection, data pre-processing, model building and model validation. Here NIR measurements are correlated to properties like lignin content.

Soft sensors also can be built with other methods like using ANN, Artificial Neural nets. There are advantages and disadvantages with the different methods, but also commonalities. You need good data for building the models. This means that data need to be spread out in the value space in a good way. If we only have “white noise” the models will be unusable. We need to vary all variables in a systematic way to get useful data for model building.

2.3 Gaussian process regression model

Gaussian Process Regression takes more memory but gives better regression models than many other methods like (Nonlinear) System Identification, Neural Networks and Adaptive learning models. Can also be Combine with physics-based models. The method is presented in e.g. Fredrik et al. [14]. In **Figure 6** we see a first attempt to predict kappa number of pulps after a digester for two different wood

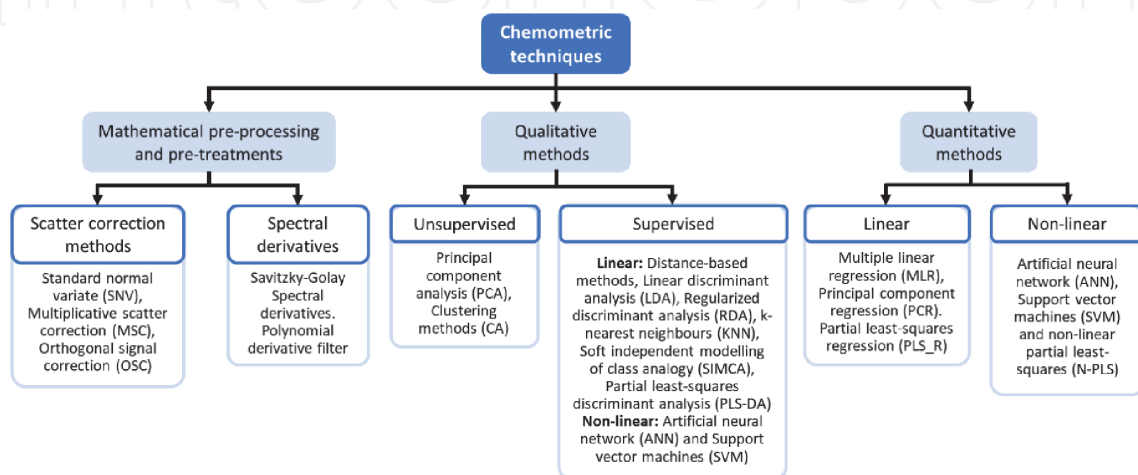


Figure 4.

A number of methods that can be used to develop soft sensor models from process data.

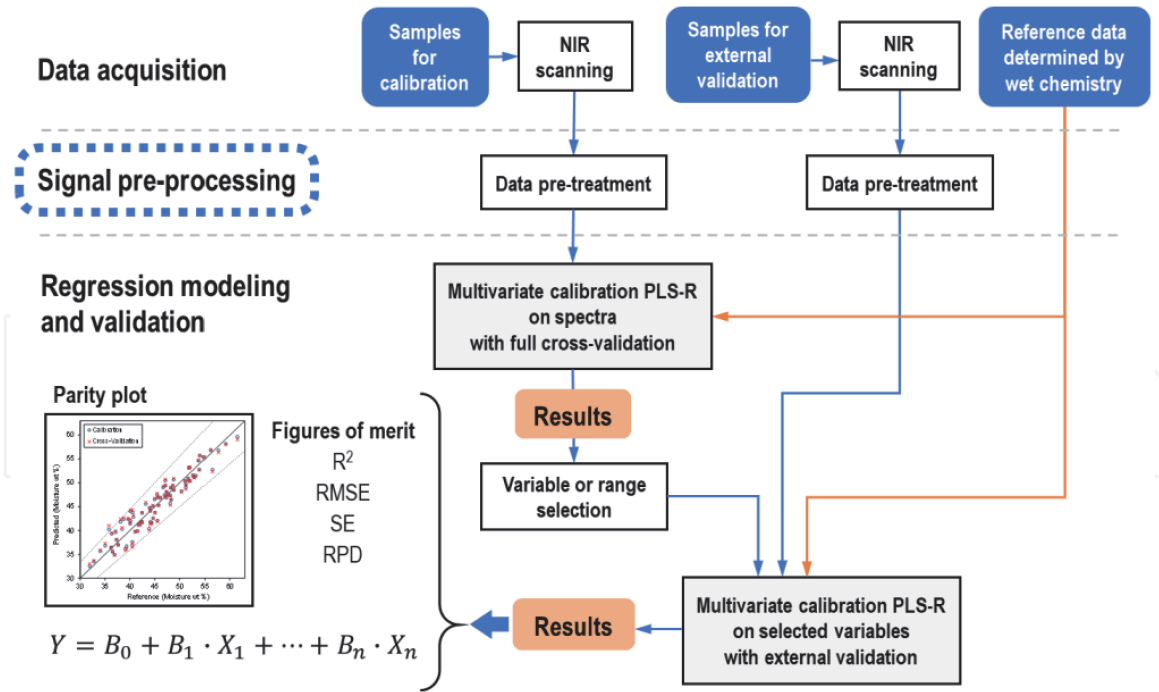


Figure 5. Data flow for building and verification of soft sensors.

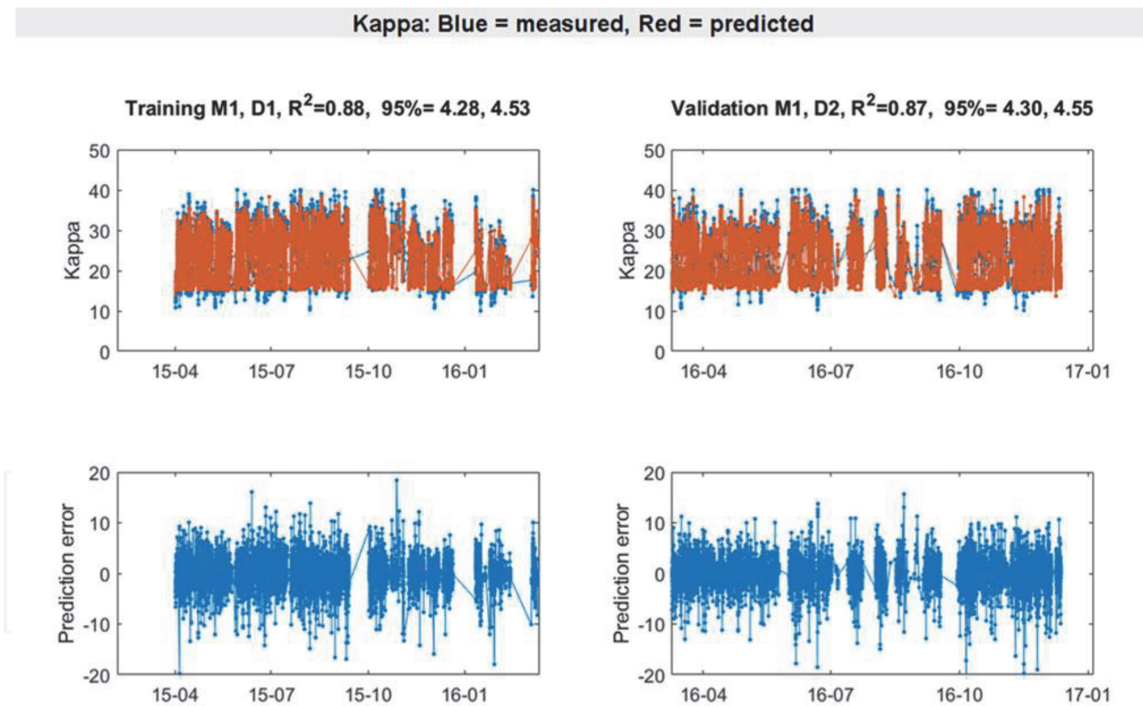


Figure 6. Example of Gaussian process regression (GPR) for kappa prediction.

types, hardwood and soft wood. The training data fits quite well, while the predictions are less good. By using more data and fine-tune the estimation of residence time in the reactor the prediction power became significantly better. It went from $R^2 = 54$ to $R^2 > 90$.

2.4 Artificial neural nets, ANN

Artificial neural nets try to mimic the brain. In a simple way we can use the equation below to show how it is calculated:

$$\hat{y}(t|\theta) = a_1 * \kappa(\gamma_1 + \beta_{11}\varphi_1(t) + \beta_{21}\varphi_2(t)) + \alpha_2 * \kappa(\gamma_2 + \beta_{12}\varphi_1(t) + \beta_{22}\varphi_2(t)) \quad (4)$$

In **Figure 7** we see three input variables to the left. Each variable is multiplied with a weight factor towards the two summa-nodes, where the products are summarized. Next these values can be treated to pass a threshold or only be passed on and multiplied with a second constant α_i . The two products are summarized again, and we get a prediction of the value of a wanted property. When you build the net, you look at the difference between the measured and the predicted value and adjust the weight factors until you get a good fit. When you have been testing one set of input variables you go to the next and proceed for all data you have and try to get a fit that is the best for all input variables together. This is a simple net with only one “hidden layer”, but you can have much more complex versions with many variables and many layers. If you have many layers the problem though can be that you get a good fit for the training data but it may also give risk for “over-fitting”, which means less stable predictions.

An example of a first commercial application of ANN was for prediction of NOx in power plants. In **Figure 8** below we see a regression for the power boiler number four in Vasteras.

2.5 PLS, partial least square regression and factorial design of experiments

PLS is very popular to use for making prediction models after performing factorial designs of experiments. The basic idea is to start with a linear regression for a line, $y = a + b * x$, and adding non-linearity by $+c * x_2$ and if there are more than one variable the interaction between variable 1 and 2 by $d * x_1 * x_2$. The polynomial for a property like a strength property of a paper then becomes

$$y = A + B * x_1 + C * x_2 + D * x_1^2 + E * x_2^2 + F * x_1 * x_2 \quad (5)$$

Here A-F are constants you get from fitting the experimental data to the model. If we use factorial design, it means that we try to expand the prediction space as much as possible within given borders. This means that we shall have a good distribution of experimental data in all parts of the space, and not only close to origo or in one part of the space. This means for example that you shall not make correlation for one variable at a time but vary all variables in a systematic way. In Ferreira et al. [15] the Box–Behnken design is described more in detail. In **Table 2** below we see an example for three variables:

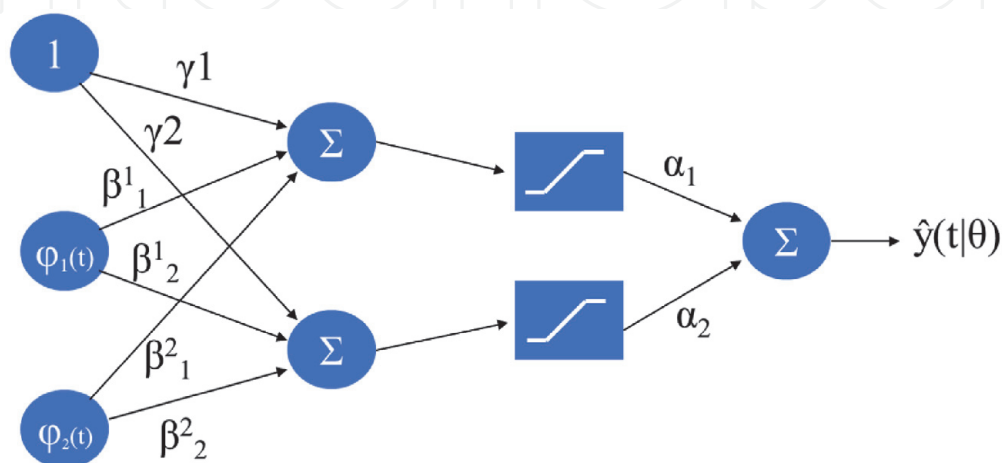


Figure 7.
A simple artificial neural net, ANN.

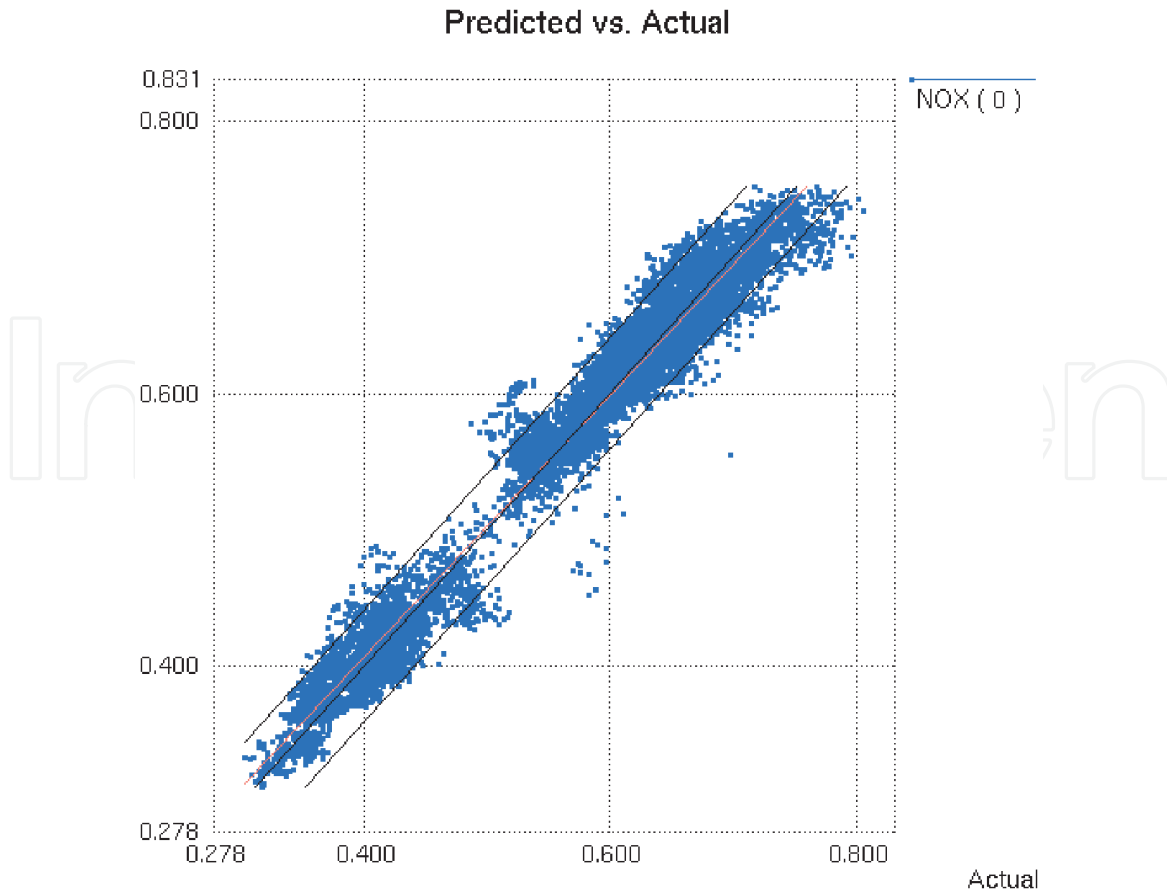


Figure 8.
 A plot showing the correlation between prediction with an ANN and measurements of the actual NOx content in the exhaust gases from a power plant (coal fired boiler 4 at Malarenergi).

Experiment no	x_1	x_2	x_3
1	+	+	+
2	+	+	—
3	+	—	+
4	+	—	—
5	—	+	+
6	—	+	—
7	—	—	+
8	—	—	—
9	0	0	0
10	$\sqrt{3}$	0	0
11	0	$\sqrt{3}$	0
12	0	0	$0\sqrt{3}$

Table 2.
 Factorial design of experiments with three important variables to predict a certain qualitative variable like paper property, lignin content, content of different plastics etc.

The first 8 experiments give the linear regression while the last four gives the non-linear components. As we vary all variables independently, we get the interaction between the variables directly. (+) means here a higher amount or

concentration of the variable while (–) means a low. (0) is Origo and $\sqrt{3}$ is where a sphere is cutting the axis.

It is important to have an equal distribution in the whole sample volume of measurements. If a high concentration of samples around origo – the impact of the “real” samples will be too small. It is better to have a few good samples well distributed instead of many around origo or some other part of the space. By varying several variables at simultaneous also catches interactions between the variables. The reason while sometimes models built from only on-line data in a plant may have very little prediction power is if we have a number of important variables with controllers, and only get the white noise due to poor control. By really varying these variables in a systematic way as proposed by factorial design, we can build robust prediction models. If the models still are not that good, it may be because we are not varying or measuring all important variables. Then we should change the variables in the factorial design. If you do not know which variables are the most important you can start with the factorial design scheme in **Table 2** but add more variables and just vary them around origo and perhaps some other random point. From this first scan we can decide which variable to focus more experiments on.

The factorial design scheme can also be seen as values at the corners of a cube and where the axis crosses a sphere around the cube as seen in **Figure 9** below:

If it is expensive to run all experiments, you can make a reduced factorial design, where you principally pick some of the variants randomly and make a PLS model. You then add one or two experiments and see how much better it becomes and proceed until you feel satisfied. This can be illustrated as in **Figure 10**.

Principally the regression is made so that you start with a line through all data in the space and calculate the square of the distance between the point and the line. You add all values for all points. Then you change the direction and make a new try. This then proceeds until you have found a line that has least sum of square errors. You then make an axis perpendicular to this first line and proceed to find a plane.

One example can be seen in **Figure 11**.

$$\begin{aligned} \text{Strength} = & A + B * \text{concentration of filler} + C * \text{ration_longfiber_to_shortfiber} \\ & + D * (\text{concentration_of_filler})^2 + E * (\text{ration_longfiber_to_shortfiber})^2 \\ & + F * (\text{ration_longfiber_to_shortfiber} * (\text{concentration_of_filler})) \end{aligned} \quad (6)$$

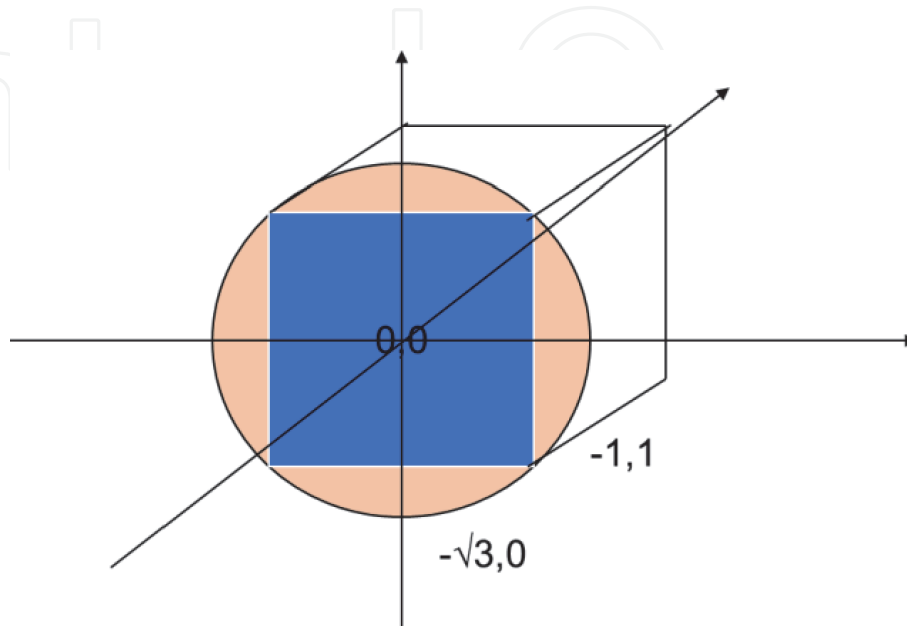


Figure 9.
Factorial design with values in all corners of the cube and where axis cross a sphere surrounding the cube.

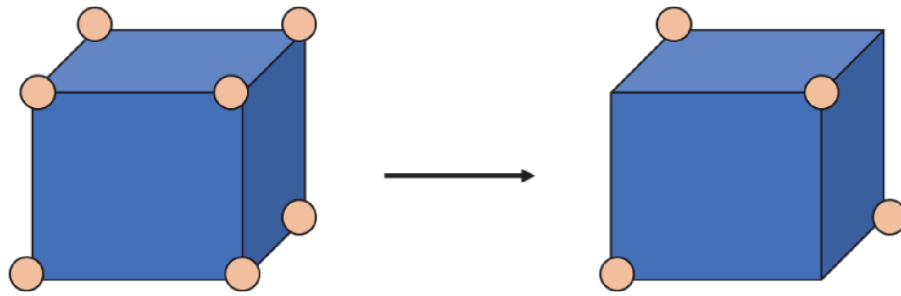


Figure 10.
Reduced factorial design.

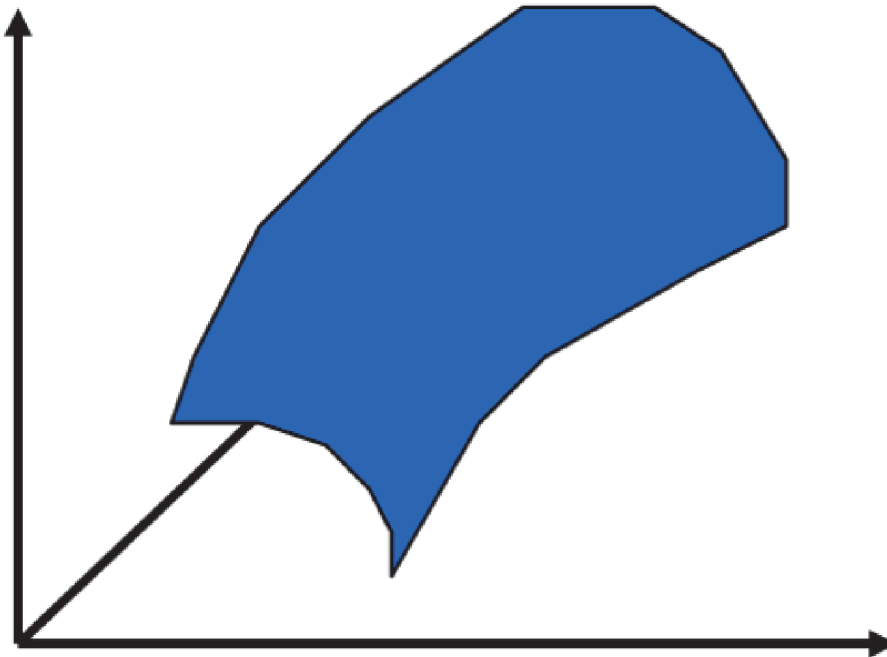


Figure 11.
The plane direction is corresponding to the line, the down wards bending the non-linearity and the cross bending of the surface shows interaction between the different variables x_1 , x_2 and x_3 .

In **Figure 12** we see what wavelengths have importance and to what degree for predicting the investigated property. At the top we have regression coefficients for AIL, Acid Insoluble Lignin, and at the bottom for ASL, Acid Soluble Lignin.

We can see from the regression coefficients in **Figure 12** that there is a significant difference between the spectra, indicating that the chemistry differs quite a lot. This as each wavelength corresponds to vibrations of a certain chemical bonding, like C-H, C-H₂, C-O, C=O, etc. This example is taken from Skvaril Jan [16].

Confounding means that some effects cannot be studied independently of each other. This is very much the case in combustion processes, water treatment, process industries like pulp and paper etc.! This is why the factorial design of experiments make so much sense. In some cases, though there is no interaction between different variables, and then it might be OK to build linear models, but this is often more exceptions than the rule. There are a number of PLS methods. One popular version is PLS Regression which is presented by e.g. Svante et al. [17].

2.6 Fault diagnostics

It is interesting to determine both process and sensor faults. This can be performed in many different ways. You can listen to noise from an engine that

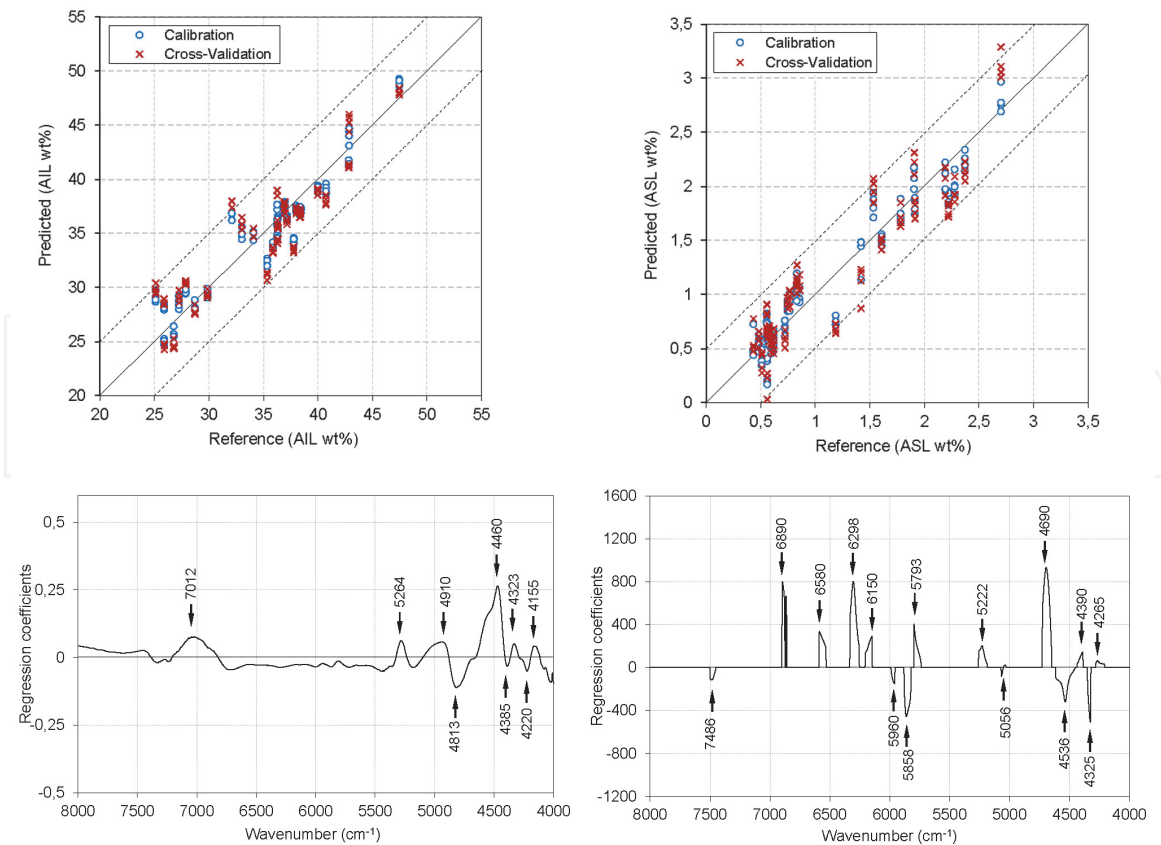


Figure 12.
Example of regression between wave lengths and lignin concentration in wood.

indicates some fault. Or you measure that the temperature has become too high somewhere. Fault detection can be systemized by using different tools and BN, Bayesian Networks, is a tool suitable for identifying causality relations and probability for different type of faults simultaneously.

2.6.1 Bayesian networks (BN)

Bayes was a priest in Scotland first discussing correlation versus causality. Correlation means that you can see how different variable are connected to each other, while causality means to take it a step further and also identify true dependence between a variable and a fault or similar. If we see that there is a correlation between homeopathic levels of a substance and effect on health, this can be a correlation but hardly that the homeopathic medicine is causing the good health. A lot of correlations are just random! With the Bayesian net you try to find the causality between different variables and a fault or similar and also quantify this. If we have a lot of experimental data we can use this to tune the BN, but if we do not have it but know from experience that there is a causality, we can make a reasonable guess of the importance in relation to other variables and use this for the BN. This gives an opportunity to make prediction models without “big data” and you can combine this input with real measurements in the plant.

Applications of BN for condition monitoring, root cause analysis (RCA) and decision support has been presented in e.g. Weidl G., Madsen A L, Dahlquist E [18]; [19, 20] and adaptive RCA in Weidl et al. [21]. Weidl and Dahlquist [22] also has given a number of examples of RCA in pulp and paper industry applications like digesters and screens. In Weidl and Dahlquist [23] applications more generally for complex process operations are presented where object-oriented BN are utilized.

In Widarsson [24] Bayesian Network for Decision Support on Soot Blowing Superheaters in a Biomass Fuelled Boiler was presented and in.

If we have a number of BN variables $U = \{A_i\}$ and parent variables $pa(A_i)$ of A_i we can use the chain rule for Bayesian networks to give the probability for all variables A_i as the product of all conditional probability tables (CTP) $P(S_k|H_1, H_2, \dots, H_n)$. Here S_k is the child node which can be observed status, measured values by some meter, a trend or similar) and H_i is the parent node (assumed causes or conditions causing a change in the child node state). The CPT can be trained by real measurements with conditions and related failures or created by using experience by operators or process experts. This is of specific interest when you want to include possible faults occurring very seldom, but severe when actually happening. Data might also be created for training by running a simulator with physical models and with different faults.

The chain rule for all CTPs is as seen in Eq. 7.

$$P(U) = P(A_1, \dots, A_n) = \prod_i P(A_i|pa(A_i)) \quad (7)$$

An example of a BN for a Root Cause Analysis function for a screen in e.g. pulp and paper industry can be seen in **Figures 13** and **14**.

2.6.2 Anomaly detection

If we have identified that a variable should be within certain limits or we have made a model using SVM or PCA or similar, we can see if the measured set of variables is within the borders for a class or group. Both these types of measures can be used for anomaly detection. This can be very useful to identify if the process goes out of normal operations even if you have not passed the limits for a single variable.

2.7 Classification and clustering

2.7.1 Principal component analysis (PCA)

Svante et al. [25] have presented the tool PCA in an article already 1987. PCA is often in the same software package as PLS but has a different use. In the PCA we

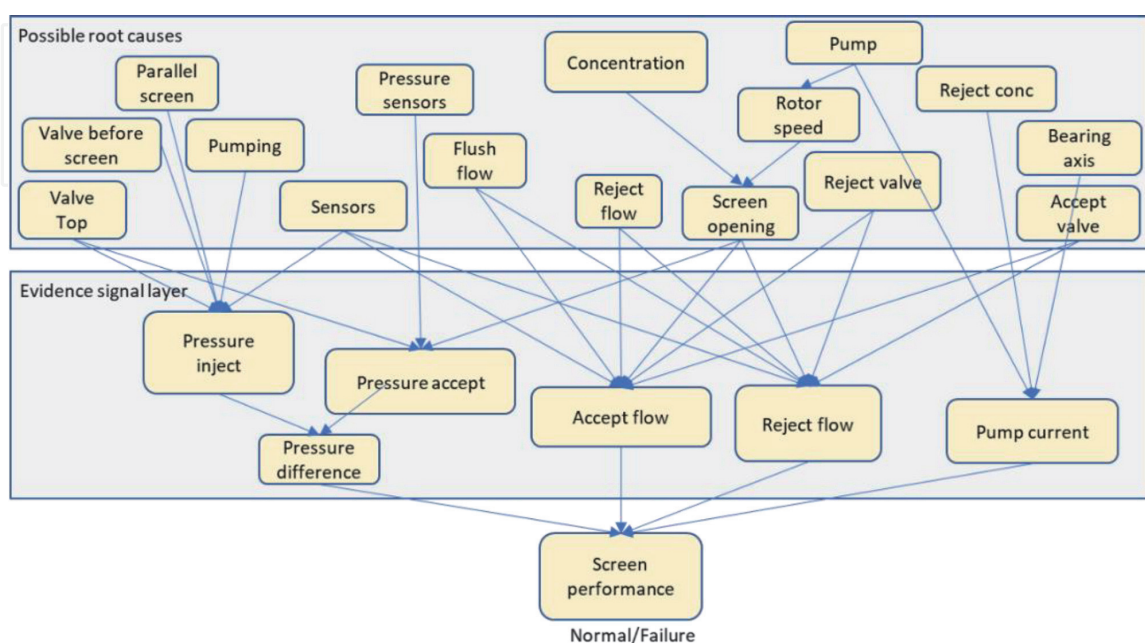


Figure 13.
 A Bayesian model for RCA (root cause analysis) of a screen.

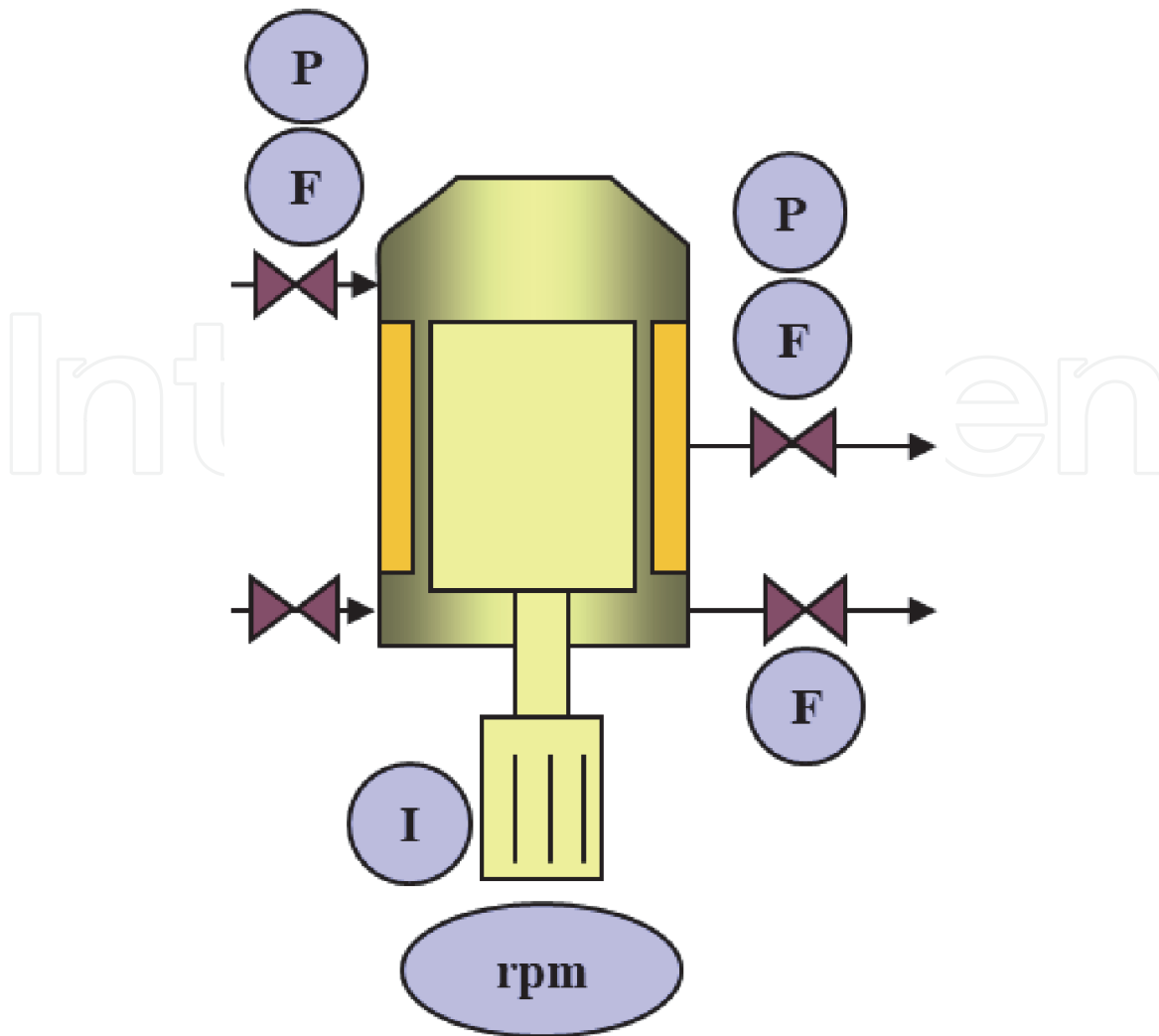


Figure 14.
A principal drawing of a screen with sensors.

plot all measured data onto different planes to see how the variables distribute in the plane. From this we can see that variables close to each other are affecting a certain property in the same way, while those on the opposite side of the diagram are having also the opposite effect. If the variables are close to Origo, we can believe they have not much effect at all on the studied property.

The score vector is a column of T . There will be one score vector for each single PC. Each experiment will have one value in the PC1 and one in the PC2. You plot all experiments in a coordinate system with PC1 and PC2. If we plot all experiments in a diagram with PC1 and PC2 we can get as in **Figure 15**.

In **Figure 15** we have plotted the time series of measurements and can see that there is a development from left to right as time passes by, along PC1. This shows that something is happening by time. We can also make a loading (p) plot. The loading plot shows how much each variable contributes to each PC. Each PC can be seen as the linear combination of the original variables

$$PC_i = \sum p_{ji} x_i \quad (8)$$

The loadings are the coefficients p_{ji} . Each variable can contribute to more than one PC. If we have more than two PCs, it can contribute to all PCs. In **Figure 16** we see the p-plot for a number of variables:

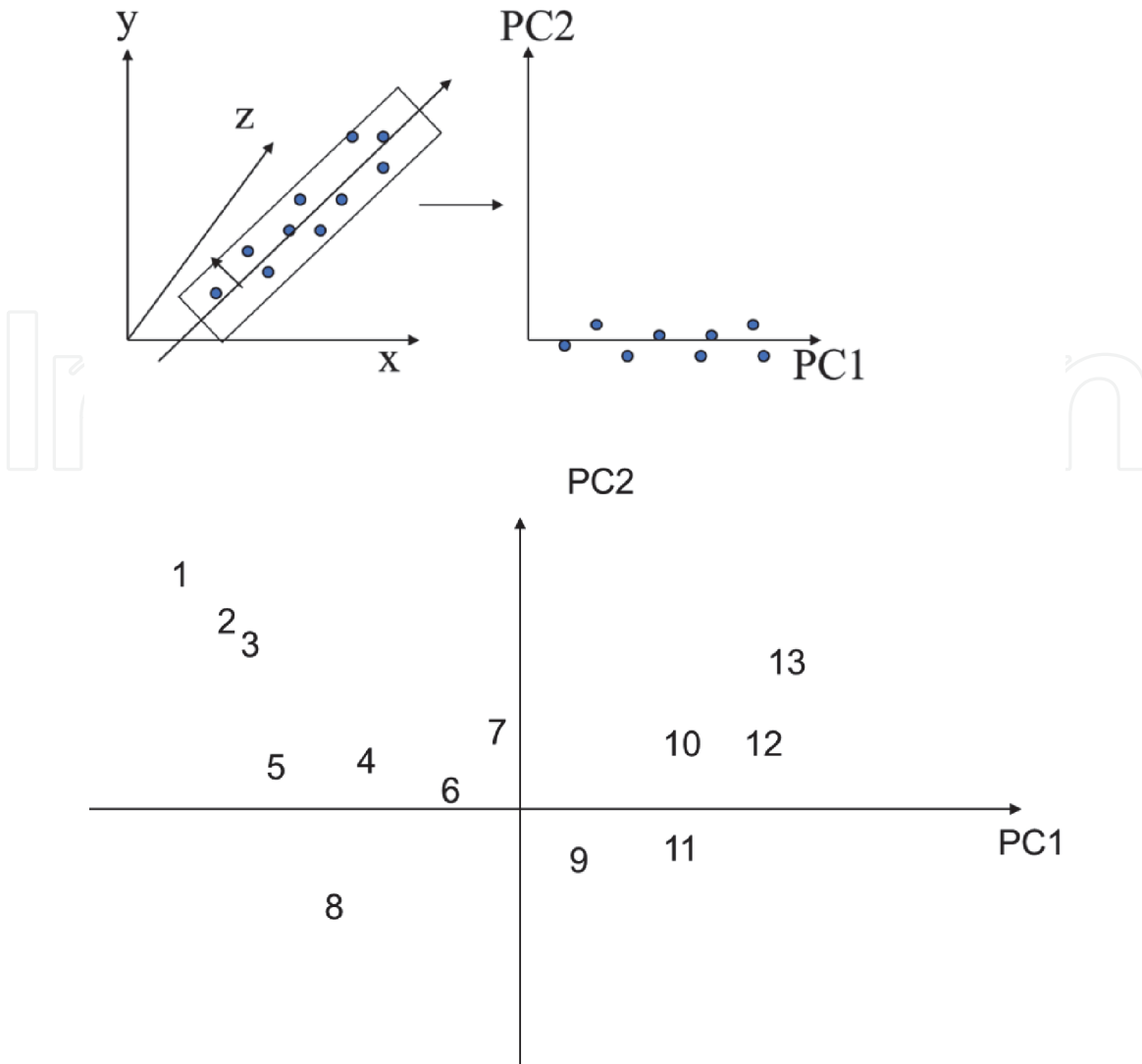


Figure 15. Score plot (t). First sample no 1 at $t = 0$ and following nos: following time steps.

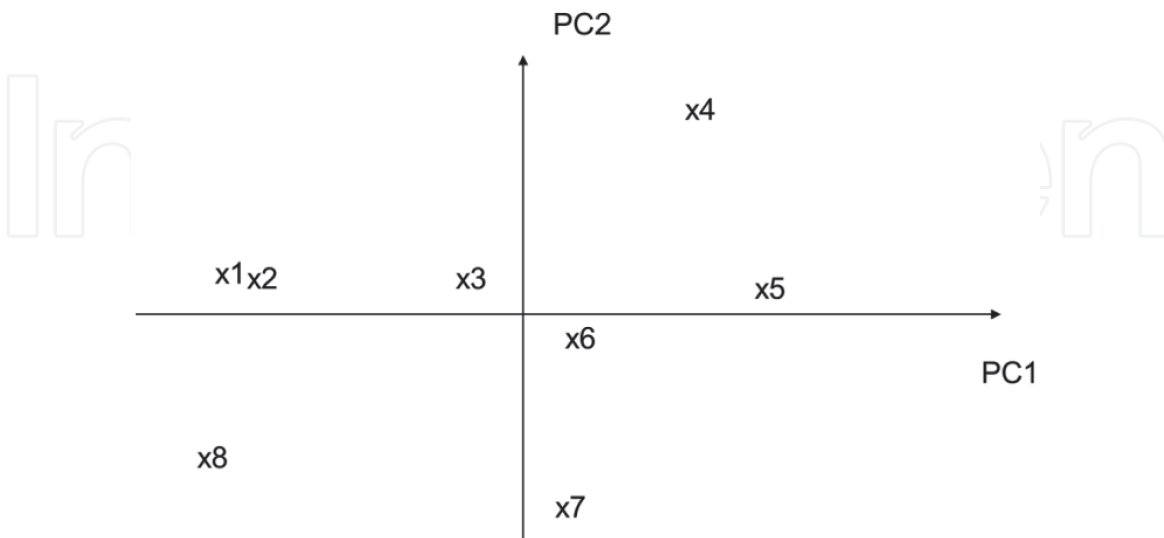


Figure 16. P-plot for eight variables in the PC1 – PC2 coordinate system.

From **Figure 16** we can see that X3 and X6 have small impact while X4 and X8 have stronger impact but reverse to each other. X1 and X2 are following each other closely.

In **Figure 17** it can be seen that when the set of variables is within the circle the process is running OK, but when going outside you should take a look and try to get it inside the circle again. This is a bit towards diffuse control, like in the human body.

You can use the p-plot also to classify a number of faults. In **Figure 18** we see an example where vibrations, temperatures and electric power consumption was used to predict different type of faults. The faults were implemented at the lab and the variables measured. From this we could see that the variables were forming different patterns.

The PLS is principally partial least squares or projection to latent structures. Principally you do an interactive PCA for both X and Y matrices.

$$X = \sum T * PT + E \quad (9)$$

$$Y = \sum U * QT + F \quad (10)$$

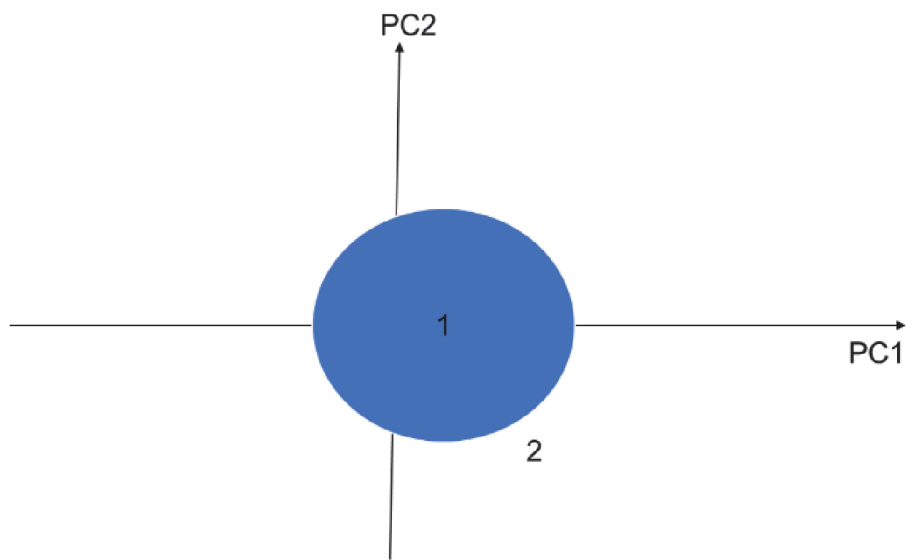


Figure 17.
Using the plot to control the process by keeping within a certain area of the PC1-PC2 space.

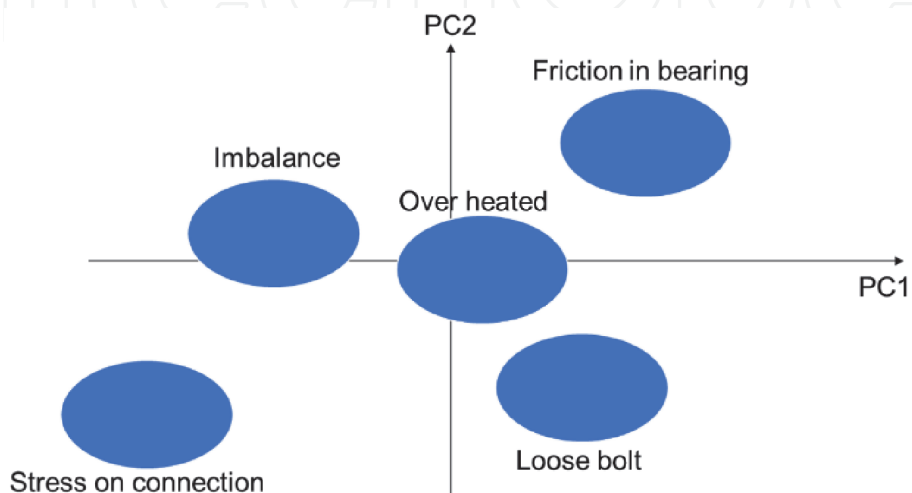


Figure 18.
Use of plots to classify different faults.

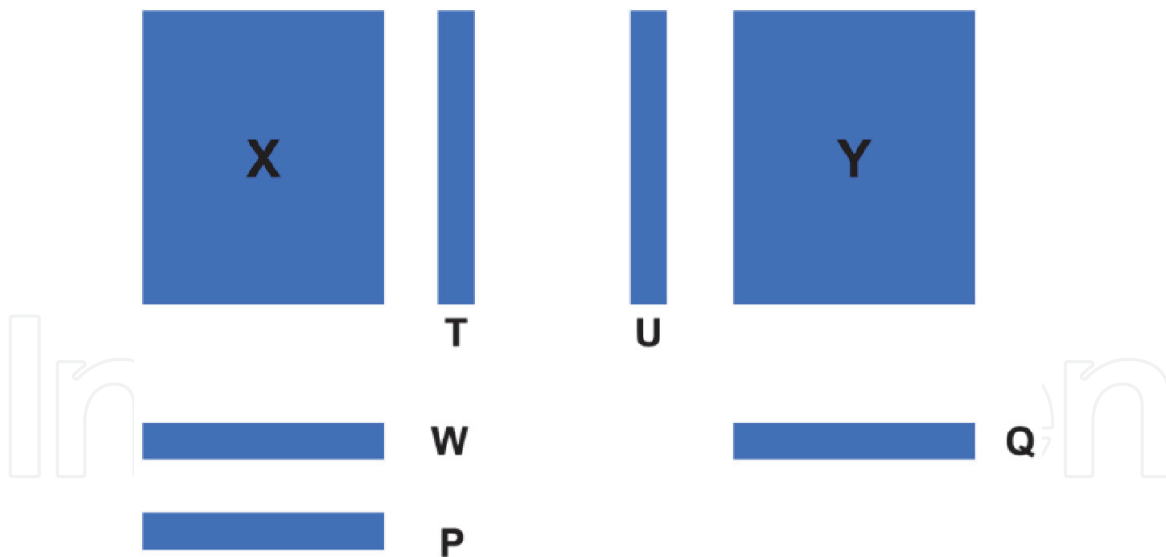


Figure 19.
 The principles for PLS (partial least square) regression.

This can schematically be seen in **Figure 19** below.

U gives starting values for T, and T back to U iteratively. Interdependency. When the difference between two iterations is below a certain value, we take this as the result.

There are a number of versions of this. PLS2 general = all Y; PLS1 for each single Y; PCR also for each single Y, and no interactivity between Y and X (first X, then Y); PCR is often used by statisticians while PLS by application engineers normally.

The result from the PLS regression will be a polynom. If only linear: $Y_1 = A + Bx_1 + Cx_2$. If also nonlinear: $Y_1 = A + B * X_1 + C * X_2 + D * X_1^2 + E * X_2^2$. If also interaction between variables: $Y_1 = A + BX_1 + CX_2 + DX_1^2 + EX_2^2 + FX_1X_2$. If we have more variables than two, we add X_3, X_3^2 and interaction between X_3 and the other variables, etc. These are used for prediction of Y_1 . If you want to study several quality aspects using the same experiments, you add polynoms for Y_2, Y_3, Y_i in the same way, but with different constants of course.

2.7.2 Support vector machines (SVM)

In SVM we try to find the balancing point for different clusters and then try to distribute the different measured values as close as possible to one of these cluster balancing points. This is giving a similar type of clustering but is usually used for a big set of data where you want to find out how many clusters there might be. You can systematically test to have more or less clusters and see how the data fits from a statistical perspective into more or less clusters.

2.8 Adaptive control

In Narend S. Kumpati [26] Adaptive control using neural networks is presented. Since then MPC, both “fixed” and adaptive, have come to use in many applications in process industry. There is even a Journal of Adaptive Control and Signal Processing. In a recent number, April 2020, Merve et al. [27] discuss Improving transient performance of discrete-time model reference adaptive control architectures. This area is binding AI, modeling and control together.

3. Architectural structure

In **Figure 20** the structure implemented in the FUDIPO project (www.fudipo.eu) [28] with respect to different functions is outlined. In the chapter about the data structure Tieto has addressed different programs. These two are complementary. One is a set of commercial software linked into the Tieto HMI3 platform. Examples of the commercial tools are MatLab/Simulink for mathematical calculations and simulation, Hugin for Bayesian Network configuration and Dymola for Modelica implementation for simultaneous solver simulation.

In the second structure we have primarily open source programs like Red Node for configuration of the complete system, linking everything together. MatLab is replaced by Python and Simulink with OpenModelica, Dymola and these are then complemented by other simpler software for different functions. The idea is that you can test all functionalities together in the open source environment. If you have a smaller system you can configure and use this also for “the real case”. If you have a bigger system demand you probably go for commercial software to also get support for the functions, and perhaps also make a service contract with someone who can support sustaining the system, and upgrading on a frequent basis, as the production plant is developing.

From this overview we can see that there are many possibilities with use of AI-tools, but it also takes some effort to understand which tools are useful to solve specific problems.

- The solutions must be **robust**. 100% of operational space must be covered in a reasonable way.
- Diagnostics must **detect** real faults, but **avoid** detect “false faults”
- **Autonomous** systems may be good, but you have to identify the borders and **limits** and what are important functions to work with.

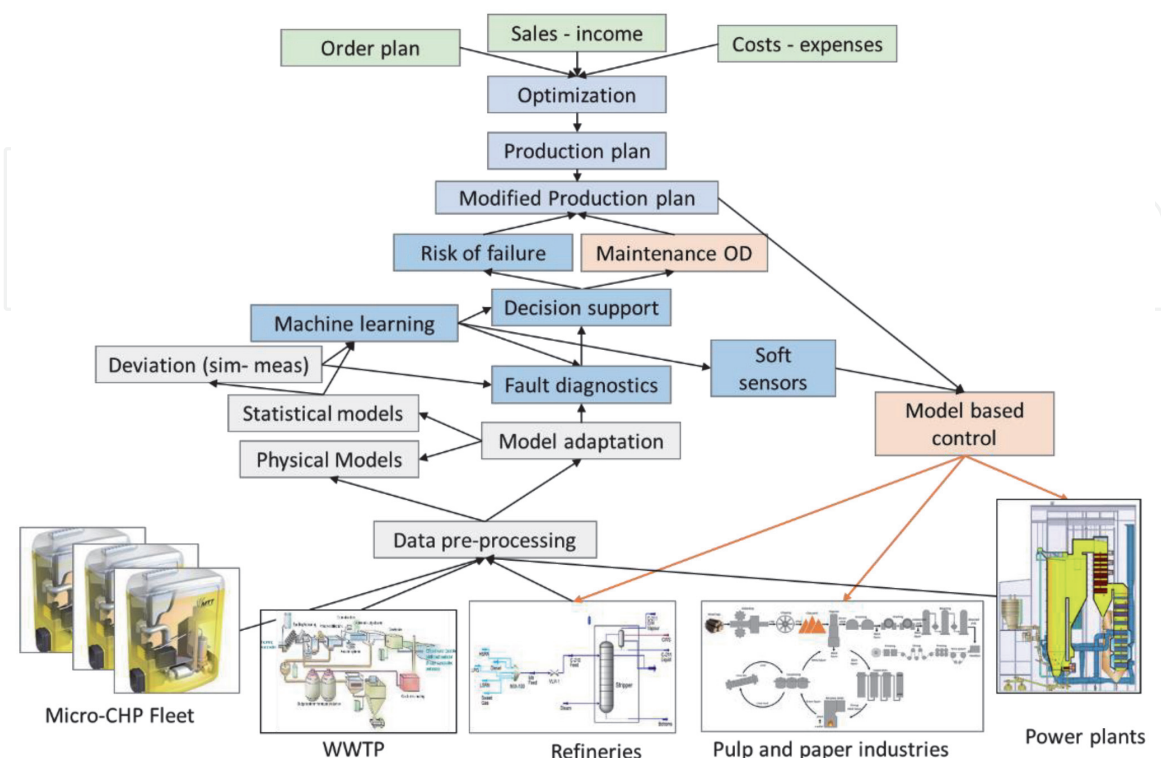


Figure 20. Layout of a complete system where different level and functions are connected and integrated.

- Need to **define the problem** to solve!
- Optimization and adaptive systems and functions should include all important functions. To do so you also need to **vary the important variables**. You cannot train a system on constant values! Factorial design of “experiments” is then important.
- Many new tools are being accessible, but you **need to understand** how they work! Do not guess.

Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 723523.




Author details

Erik Dahlquist*, Moksadur Rahman, Jan Skvaril and Konstantinos Kyprianidis
Malardalen University, Vasteras, Sweden

*Address all correspondence to: erik.dahlquist@mdh.se

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. Distributed under the terms of the Creative Commons Attribution - NonCommercial 4.0 License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited. 

References

- [1] Mosavi, A., Salimi, M., Faizollahzadeh Ardabili, S., Rabczuk, T., Shamshirband, S., & Varkonyi-Koczy, A. R. (2019). State of the art of machine learning models in energy systems, a systematic review. *Energies*, 12(7), 1301.
- [2] Correia, F. M., d'Angelo, J. V. H., Almeida, G. M., & Mingoti, S. A. (2018). Predicting kappa number in a Kraft pulp continuous digester: A comparison of forecasting methods. *Brazilian Journal of Chemical Engineering*, 35(3), 1081–1094.
- [3] Dahlquist E. Editor: Book "Use of process simulation in pulp and paper industry. Published by EU. Product of COST E 36. May 2008a. ISBN ISBN 978-91-977493-0-5
- [4] Dahlquist, Erik (2008b) "Process simulation for pulp and paper industries: Current practice and future trend," Review Paper after Invitation. *Chemical Product and Process Modeling: Vol. 3 : Iss. 1, Article 18*. Available Open Source at: <http://www.bepress.com/cppm/vol3/iss1/18>
- [5] Phatwong, A., & Koolpiruck, D. (2019, July). Kappa Number Prediction of Pulp Digester Using LSTM Neural Network. In 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 151–154). IEEE.
- [6] Abbas, S., Khan, M. A., Falcon-Morales, L. E., Rehman, A., Saeed, Y., Zareei, M., & Mohamed, E. M. (2020). Modeling, simulation and optimization of power plant energy sustainability for IoT enabled smart cities empowered with deep extreme learning machine. *IEEE Access*, 8, 39982–39997.
- [7] Karlsson, Christer P.; Avelin, Anders; and Dahlquist, Erik (2009) "New methods for adaptation to degeneration in process models for process industries," *Chemical Product and Process Modeling: Vol. 4 : Iss. 1, Article 25*. DOI: 10.2202/1934-2659.1127. Available Open Source at: <http://www.bepress.com/cppm/vol4/iss1/25>
- [8] Lorencin, I., Andelic, N., Mrzljak, V., & Car, Z. (2019). Genetic algorithm approach to design of multi-layer perceptron for combined cycle power plant electrical power output estimation. *Energies*, 12(22), 4352.
- [9] Karlsson Christer, Anders Avelin, Erik Dahlquist.:How to make better use of all the process data collected in process industry and power plants. 6th Eurosim congress on modeling and simulation, September 9-13, Ljubljana, Slovenia, 2007.
- [10] Liu, Y., & Xie, M. (2020). Rebooting data-driven soft-sensors in process industries: A review of kernel methods. *Journal of Process Control*, 89, 58–73.
- [11] Carlsson, A. Perspectives on the discovery of central monoaminergic neurotransmission. *Annual Review of Neuroscience (Palo Alto, CA)* 1987. 10. 19–40.
- [12] Gattami Ather: Reinforcement learning for multi-objective and constrained Markov decision processes. 2019. *Journal arXiv preprint arXiv: 1901.08978*.
- [13] Werbos Paul: A Menu of Design for reinforcement learning over time (p 67–95). In Miller Thomas, Sutton Richard and Werbos Paul (editors): *Neural Networks for Control*. 1990, Book ISBN 0-262-13261-3 MIT.
- [14] Lindsten Fredrik, Thomas B. Schön, Andreas Svensson, Niklas Wahlström : Probabilistic modeling – Linear

regression & Gaussian processes
February 23, 2017. Uppsala University
Press.

[15] Ferreira S.L.C, R.E. Bruns, H.S. Ferreira, G.D. Matos, J.M. David, G.C. Brandao, E.G.P. da Silva, L.A. Portugal, P.S. dos Reis, A.S. Souza, W.N.L. dos Santos (2007) Box-Behnken design: An alternative for the optimization of analytical methods. *Analytica Chimica Acta* 597, 179–186.

[16] Skvaril Jan, Konstantinos G. Kyprianidis & Erik Dahlquist: Applications of near-infrared spectroscopy (NIRS) in biomass energy conversion processes: A review. *Journal of Applied Spectroscopy Reviews*, Volume 52, 2017 - Issue 8.

[17] Wold Svante, Michael Sjostrom, Lennart Eriksson: PLS-regression: A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems* 58, 2001, 109–130. www.elsevier.com/locate/chemometrics

[18] Weidl G., Madsen A L, Dahlquist E. (2002) Bayesian networks for root cause analysis in process operation, *European Journal of Operational Research*, Special Issue on "Advances in Complex Systems Modeling".

[19] Weidl, G., Madsen, A.L. and Dahlquist, E. (2002a). "Condition Monitoring, Root Cause Analysis and Decision Support on Urgency of Actions". In Book Series FAIA (Frontiers in Artificial Intelligence and Applications), vol.87, A.Abraham et al. (Eds.), *Soft Computing Systems - Design, Management and Applications*, pp. 221–230. IOS Press, Amsterdam, the Netherlands

[20] Weidl G, Madsen A L, Dahlquist E (2002b) Condition Monitoring, Root Cause Analysis and Decision Support on Urgency of Actions, 2nd International conference on Hybrid Systems, Dec 1–4, Santiago, Chile

[21] Weidl G, Vollmar G and Dahlquist E (2003): Adaptive root cause analysis under uncertainties in industrial process operation, foundations of computer-aided process operations conference, USA, Florida, January 12–15, 2003

[22] Weidl G, Dahlquist E. (2002) Root cause analysis for pulp and paper applications, In Proceedings of 10th SPCI Control Conference, Pp 343–347, Stockholm, Sweden, June 3-5, 2002.

[23] Weidl G., Madsen A. And Dahlquist E.: Decision support on complex industrial process operations. Chapter no 18 p 313- 328 in the book *Bayesian networks, a practical guide for applications*. Editors Pourret O., Naim P. and Marcot B. John Wiley. ISBN 978–0–470–06030–8. 2008.

[24] Widarsson B, Karlsson C och Dahlquist.E: Bayesian Network for Decision Support on Soot Blowing Superheaters in a Biomass Fuelled Boiler, *PMAPS*, Sept 13–17, 2004, Baltimore, USA.

[25] Wold Svante, Esbensen Kim and Geladi Paul: Principal Component Analysis. *Chemometrics and Intelligent Laboratory Systems*, 2 (1987) 37–52

[26] Narend S. Kumpati : Adaptive control using neural networks. (). In Miller Thomas, Sutton Richard and Werbos Paul (editors): *Neural Networks for Control*. 1990, Book ISBN 0–262–13261-3 MIT, p 115–142.

[27] Dogan K. Merve, Tansel Yucelen, Wassim M. Haddad, Jonathan A. Muse: Improving transient performance of discrete-time model reference adaptive control architectures. 27 April 2020. <https://doi.org/10.1002/acs.3114>. *Journal of Adaptive Control and Signal Processing*

[28] FUDIPO (2020) Description of an open platform based on Node Red for AI use in process industry, url: www.fudipo.eu