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

5,300
Open access books available

130,000
International authors and editors

155M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

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

Chapter

A Framework for Learning System for Complex Industrial Processes

Moksadur Rahman, Amare Desalegn Fentaye,
Valentina Zaccaria, Ioanna Aslanidou, Erik Dahlquist
and Konstantinos Kyprianidis

Abstract

Due to the intense price-based global competition, rising operating cost, rapidly changing economic conditions and stringent environmental regulations, modern process and energy industries are confronting unprecedented challenges to maintain profitability. Therefore, improving the product quality and process efficiency while reducing the production cost and plant downtime are matters of utmost importance. These objectives are somewhat counteracting, and to satisfy them, optimal operation and control of the plant components are essential. Use of optimization not only improves the control and monitoring of assets, but also offers better coordination among different assets. Thus, it can lead to extensive savings in the energy and resource consumption, and consequently offer reduction in operational costs, by offering better control, diagnostics and decision support. This is one of the main driving forces behind developing new methods, tools and frameworks. In this chapter, a generic learning system architecture is presented that can be retrofitted to existing automation platforms of different industrial plants. The architecture offers flexibility and modularity, so that relevant functionalities can be selected for a specific plant on an as-needed basis. Various functionalities such as soft-sensors, outputs prediction, model adaptation, control optimization, anomaly detection, diagnostics and decision supports are discussed in detail.

Keywords: learning system, soft-sensors, model predictive control, fault detection, isolation and identification, information fusion

1. Introduction

Despite recent economic growth, industrial plants are facing tremendous local and global competition. In order to maintain long-term competitiveness, industrial plants need to optimize their operation continuously for better quality, availability, flexibility and cost. As a consequence, industrial systems are becoming more and more complex due to the increasing coupling between highly nonlinear and stochastic subsystems or sub-processes. Often these systems include many control loops and operate under multiple operational constraints. Hence, the development of new methods and tools for optimal operation, monitoring and control of complex industrial systems is a matter of utmost importance. Rapid development of industrial automation, high-performance computing, artificial intelligence, machine

learning, big data, cyber-physical systems, advance sensors, internet of things and industry 4.0, stimulated the industry-wide application of advanced methods and tools needed for optimal operation, monitoring and control. Although many advanced techniques for optimal operation, monitoring and control are already available and many more emerging day-by-day, the widespread use of these techniques within the industrial domain has been particularly limited [1–3]. There are numerous reasons identified to be accountable for the limited industry-wide application.

Although introduction of advanced automation could ensure better asset utilization, the enterprise must make sure that the newly available capacities are used effectively. Need for major infrastructure overhaul and resistance to change towards new systems that requires user's skill upgrade are two major issues that hindering the industrial application. The penetration barriers for technology niches are also quite high due to the fact that the industrial automation sector is occupied by only few multinational conglomerates. One can also blame the lack of pilot applications proving the robustness of these emerging techniques. Traditionally, advanced functionalities i.e. output prediction, optimal control, diagnostics and decision support, have been developed separately by utilizing different approaches and often with different model assumptions [4, 5]. Due to this segregated approach, the integration of different functionalities has been difficult and, consequently often neglected. However, each of these activities are closely related and cannot really be conducted individually on a isolated manner. For example, a fault in the system or a sensor failure can have a significant impact on the output prediction or control. Therefore, integration among different functionalities are essential. Due to their longevity, existing automation systems of large industrial plants mostly date from the past few decades. Often replacing these automation systems completely may not be economically viable. Hence, there is a need for an architecture that will allow easy integration of advanced functionalities with both existing and state of the art automation platforms of complex industrial systems. In order to get a structured view on industrial automation and how optimal operation, control and monitoring can leverage the benefit from such systems, a brief overview of the automation pyramid as presented in **Figure 1** can be helpful.

So what exactly is the automation pyramid? It is a graphical representation of the different technological levels of automation in a industrial plant that allow communication among different technologies within each level as well as between the different levels. The framework is defined by International society of automation (ISA) within ISA-95 that is the international standard for the integration of enterprise and control systems [6]. The first level of the pyramid, commonly referred as field level, consists of devices, sensors and actuators that are used to measure different process parameters such as flow, temperature, pressure or concentration and to manipulate different process variables via different mechanical, hydraulic, pneumatic, electrical or electronic devices. The next level, referred as control level, comprises distributed control or logical devices such as the programmable logic controller (PLC), distributed control system (DCS) or proportional–integral–derivative (PID) controller. The control level uses these control and logical devices to control or regulate the devices in the field level that actually perform the physical work. They receive inputs from all field level sensors to make decisions on what actions need to be taken by the filed level actuators to meet the predefined set-points.

An example of separation between field and control level is presented in **Figure 2**. Suppose the level of a tank need to be controlled to a predefined level in a industrial plant. A level sensor measures the level of the tank in real time and transfers this information to a PID controller. The controller adjust the position of a
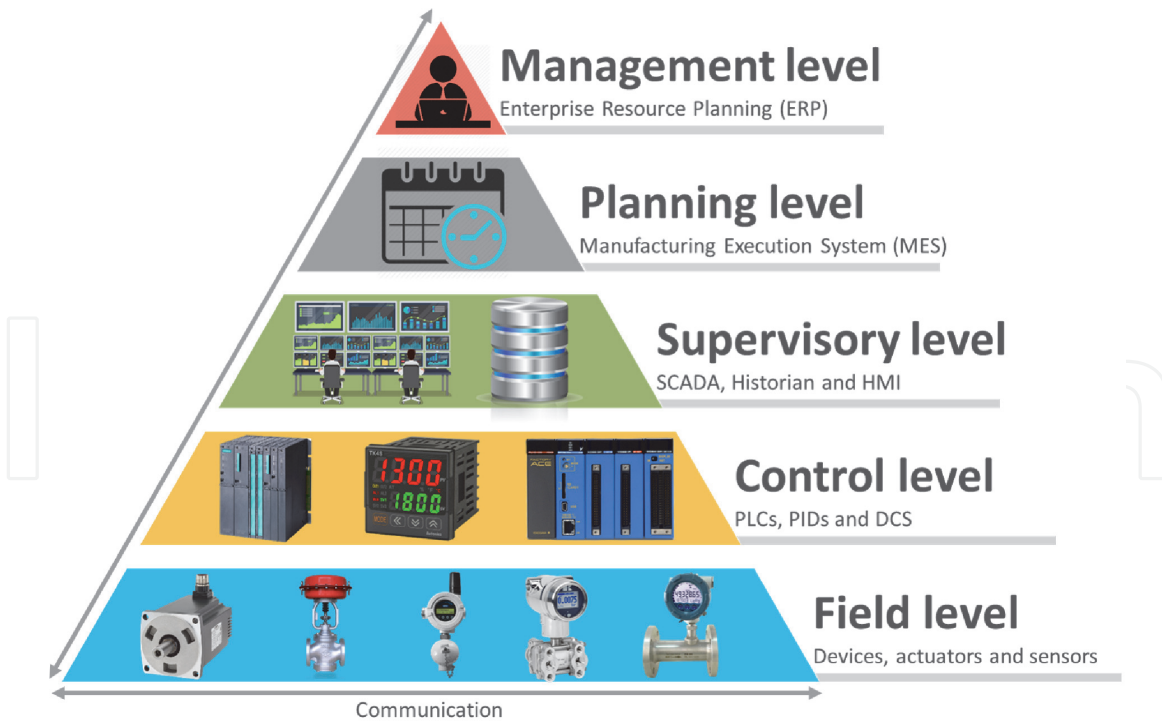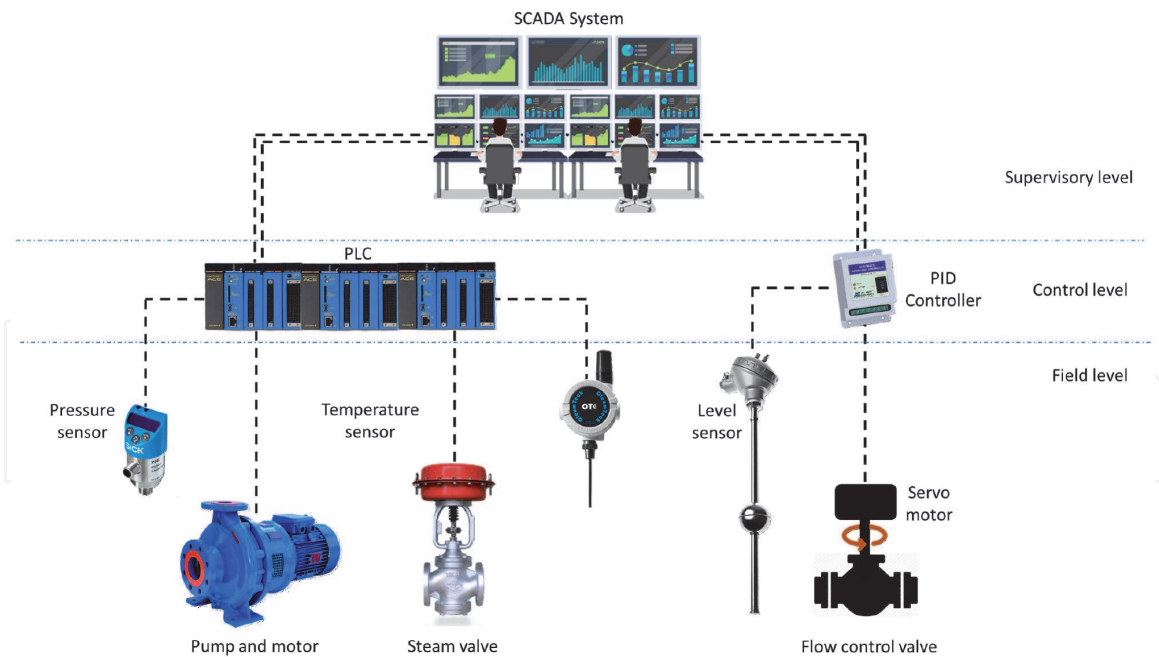
Figure 1.
The automation pyramid of a typical industrial plant.
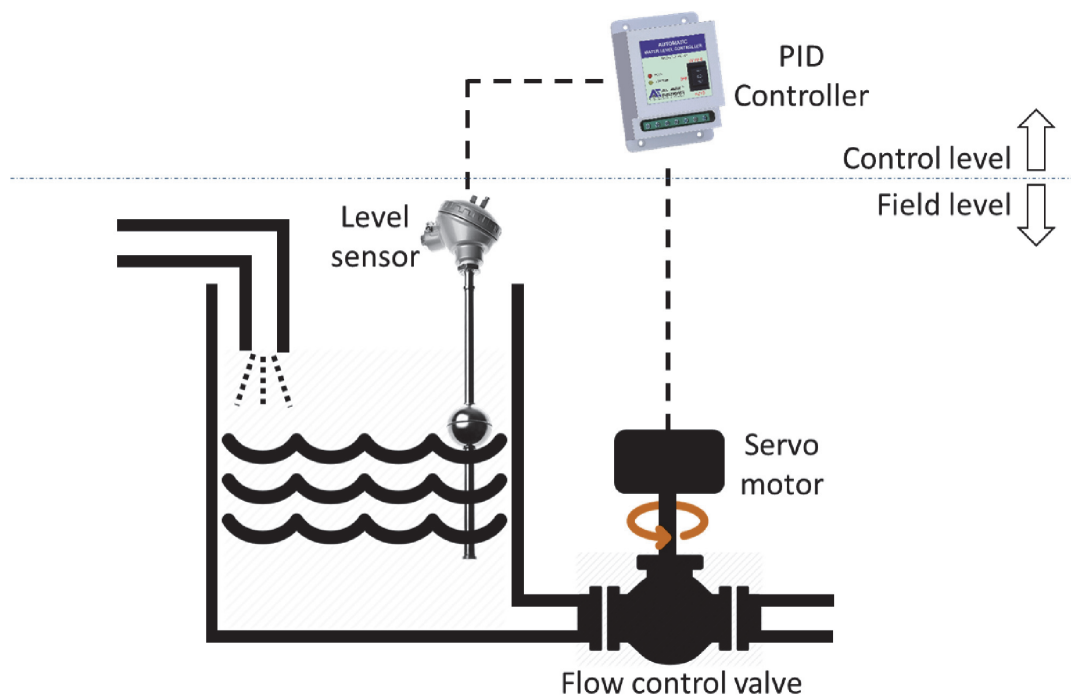
Figure 2.
Example of segregation between field and control level.

flow control valve by means of servo motor. In this scenario, the tank, level sensor, flow control valve amd servo motor belong to the field level and the PID controller belong to the control level. The supervisory control and data acquisition (SCADA) system correspond to the third or supervisory level that is used to access data and control multiple systems from a single location. The SCADA gathers information from all the subsystems and sub-processes of a industrial plant, carrying out necessary analysis and supervisory control and displaying the information in a logical and organized manner (**Figure 3**). For example, supervisory control algorithms calculate set-point values for the field level controllers (PIDs and PLCs).

Figure 3.
Relation between supervisory and control level.

Human-machine interfaces (HMI) and workstations are also included in this level. Often this level uses process historians or databases, software programs that store the historical process data. Hence, it is possible to study the patterns and find abnormalities in the processes by the experts or automated programs.

The fourth or planning level includes the manufacturing execution system (MES). MES is used to monitor the entire production process in a industrial plant from the raw materials to the finished goods. A MES performs many activities including production scheduling, management of production equipment and labor, quality control, performance analysis and maintenance management. MES provides a holistic view on the production process and allow planners to make decisions based on the available information. At the top or management level, enterprise resource planning (ERP) systems are placed to establish plant scheduling methods and material management features. ERP is a integrated software that allows organizations to monitor day-to-day business activities from manufacturing, to sales, to procurement, to accounting, to project management, to risk management, and many more. A complete ERP package typically includes enterprise performance management tool that is used to plan, budget, predict, and report on an organization's financial results. To be inline with the fourth industrial revolution widely known as industry 4.0, the structure is becoming more of a pillar than a pyramid; this enables enhanced communication beyond existing layer boundaries as well as cloud computing functionality [7]. Irrespective of its structure, advanced methods and tools can bring benefits to all levels of the automation hierarchy by providing solutions for process monitoring, coordinated process control, integrated planning and scheduling of man, machine and materials through better decision support. However, a pyramid structure is chosen here due to its simplicity and relevance.

Typically, the process components are designed to meet the operational objectives that are essential for the optimal and economic operation of the plant. Nevertheless, in reality, the process variables encounter both arbitrary and sustained deviation from their targets due to external disturbances, inherent variability and uncertainties. This is where the control system comes into play, by actively manipulating the process to ensure stable operation of the plant while keeping the product quality and specification within the target. Due to their simplicity and robustness,

more than 90% of all industrial control loops are based on PID controllers [8]. PIDs show superior performance as regulatory control of uni-variate problems, i.e. in regulating flow, temperature, pressure, level, and other variables. In principle, a PID evaluates the one-and-only process variable, decide if it is acceptable or not, and takes corrective measures if necessary. This scheme works well for control problems with only one variable or with several variables that can be manipulated independently. Despite their widespread usage, PIDs have multiple drawbacks when it comes to supervisory control of multivariate industrial processes with high level of non-linearity. Therefore, multivariate control techniques are particularly essential for supervisory control, whereas PIDs can still be used for uni-variate regulatory controls under a supervisory control loop. Different model-based and model-free multivariate process control techniques are widely studied by the research community. In particular, model-based control is widely used by the industry and has demonstrated an excellent track record [9]. However, advanced control concepts that depend on process models to maneuver the plant are prone to slow deterioration. Hence, model adaptation over time is essential to ensure optimal control of the plant.

Apart from a robust control scheme, fault diagnostics also have an important role in ensuring the optimal operation of a plant. In particular, soft faults and slow deterioration of process components over time reduce the nominal production capacity of a plant. It is often difficult to detect such faults just by looking at the process variables, and they frequently remain unnoticed until the problems become severe or lead to an unwanted plant shutdown due to component breakdown. These faults and deterioration can also affect the control system negatively and disturb the process stability.

A fault diagnostics system can be beneficial for a processing and energy plant in numerous ways. Early detection of process, equipment or component faults or deterioration can provide decision support for operators, engineers and managers at different levels, i.e. DCS, computerized maintenance management system (CMMS), MES and ERP. As a result, the operation of the plant, along with maintenance, production and inventory planning can be improved. For example, an early indication of a developing fault can provide decision support by initiating one or more suggested actions that the control system or plant operator can perform to prevent the fault development. If prevention is not possible, then early detection of such deterioration can provide an indication of the remaining useful life (RUL) of the affected component that, in turn, can provide an indication of when maintenance is needed. Once a maintenance action is planned, that can initiate procurement of the required spare parts and adjustment of the production plan based on necessity.

To achieve such cross-platform functionality, there is a need for an integrated framework for optimal control, diagnostics and decision support for the complex industrial systems. The framework needs to be generic enough to accommodate different systems with different levels of complexity. This is also necessary to cover the broad range of systems that can utilize such a framework, starting from single or multiple assets within a plant to a large fleet of assets spread over a large geographical area.

2. Framework for generic learning system

For better resource utilization, product quality and process efficiency, supervisory system of a modern industrial plant need to perform various activities including outputs prediction, model adaptation, control optimization, anomaly detection,

diagnostics and decision support. In order to enable the supervisory system to perform all these activities efficiently, we propose a framework for the generic learning system that can be integrated to the supervisory system of the complex industrial plants. The architecture is very flexible and modular, so that relevant functionalities can be selected for a particular case study on a plug and play basis.

The framework is developed in a way so that it can be retrofitted to the existing automation platforms of a complex industrial plant. This provides solution to one of the major barrier that hindering the widespread use of modern techniques emerging for optimal operation, control and monitoring of complex industrial plants. Since the framework allows easy integration of the learning system to the existing automation platforms, the need for extensive infrastructural modification and skill development reduces drastically. The overall framework for the learning system is presented in **Figure 4**. The learning system is placed in the supervisory level of the automation pyramid. However, it actively supports decision making in both planning and management level. The learning system need process data as inputs to perform systematic computational analysis. The data are gathered from the process historian or the database. The first step before performing any analysis is data assurance that includes outlier removal and noise reduction by means of various data filtering techniques. Subsequently, different advanced analysis are performed on the data ans the results are written back to the process historian. Firstly, trend analyses are carried out on the data to identify any patterns in the process parameters. Important process outputs are predicted by using physics-based and data-driven process models. Advanced control optimisation techniques are applied to calculate most optimal set-points for the low level regulatory controllers. Different physics-based and data-driven anomaly detection and diagnostics algorithms are also applied in order to find process abnormalities and faults. As a final step, results from all these analysis are used to provide robust decision support with the help of information fusion techniques. Moreover, the architecture allows integration of state-of-the-art sensors for measuring feedstock properties, different process parameters that are needed to better operation and control of complex industrial processes. Human–machine interface (HMI) are also provided for the visualization and further analysis. This is a key part of the framework that the users i.e. operators, engineers and managers will directly interact with. Hence, the HMI need to be designed such a way that it is user friendly and useful for them. This will determine
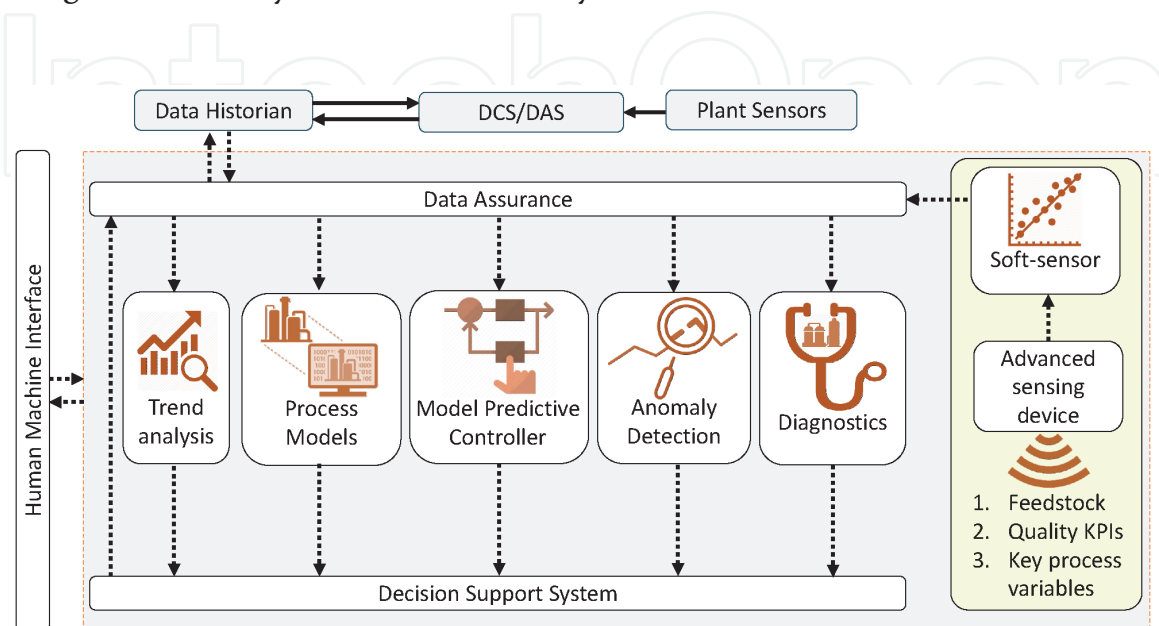
Figure 4.
*Integrated framework for the learning system of complex industrial systems.*

if the learning system will actually be used or not. Hence, the user need to be involved in the process of designing the HMI.

The different modules of the framework is discussed in the following sections.

2.1 Data assurance

Data assurance refers to different data preprocessing techniques that ensures accurate, reliable and meaningful analysis. The data preprocessing steps typically includes data cleaning, smoothing, scaling and grouping or binning [10]. Data cleaning particularly refers to detection and removal or replacement of outliers and missing data. Data smoothing on the other hand refers to removing noise from the data. Here in the data assurance layer, outliers in the data are detected and different noise reduction techniques are applied to refine the data. So what exactly meant by outliers in the data? An outlier is a measurement that differs significantly from other measurements in a dataset. The definition is quite broad in nature, allowing the analyst to decide on the boundaries that separate measurements to be considered as outliers from normal. Typically, outliers represent only a small fraction of the data and they do not follow the inner relationships present among different process variables. Very simple example of a outlier in a dataset is shown in **Figure 5**.

There are many readily available techniques that can be used for outlier detection. As each dataset is different, there are no common methods that can be applicable to every dataset. Rather, an analyst or domain expert, must examine the raw measurements and decide whether a value is an outlier or not and what methods can be used to detect it. Typically, statistical methods that are widely used for detecting outliers corresponding to significantly extreme values are mean and standard deviation, and median absolute deviation method. According to the mean and standard deviation method, a measurement is labeled as outlier if it more than three standard deviation away from the mean value. However, as both the mean and the standard deviation are sensitive to outliers, this method can be problematic in some cases. A rule of thumb is that for normally distributed dataset, mean and the standard deviation is a better choice. However, if dataset is not normal, the median absolute deviation can be used. In this case, absolute deviation from the median
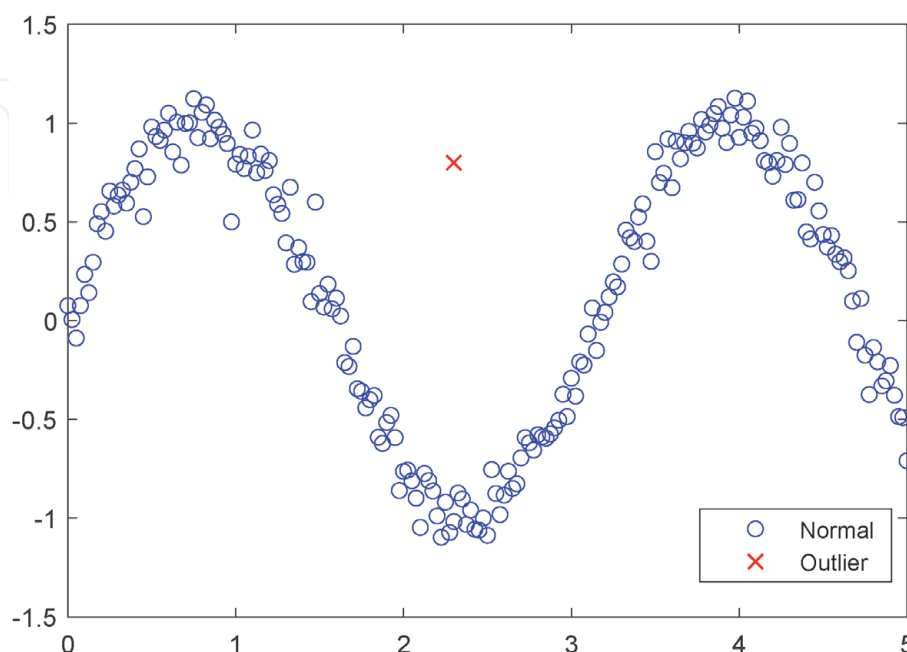
Figure 5.
Example of a outlier in dataset.

value is used instead. Normally, historical data or a window width is used to apply such techniques for time-series sensor readings.

Process data are subjected to noise. Hence, different noise reduction techniques are needed before performing different analysis on the data. A typical example of noisy sensor data and output data after smoothing is presented in **Figure 6**. However, one need to be careful when applying different noise reduction techniques. Too much data smoothing can filter out many useful information in data that can be important for different data analysis techniques. For noise reduction, time domain filters i.e. moving average filter, moving median filter, Savitzky–Golay filter, artificial neural network (ANN) and local regression smoothing, and frequency domain filters i.e. low pass, high pass and band pass filter are well known data smoothing techniques. Among these, moving median filter is simple but most powerful data smoothing technique. It particularly useful for eliminating unwanted noise from the time-series sensor data. Two of its main advantages are (a) median filtering preserves sharp edges and (b) it is very efficient for smoothing of spiky noise. However, presence of outlier in the data can effect the outcome of a moving average filter. Hence, such smoothing techniques need to be used in addition to the outlier removal step. The mathematical expression for moving average filter is presented in Eq. (1).

$$\overline{y_n} = median(x_{n-k}, \ldots \ldots, x_n, \ldots \ldots, x_{n+k}),$$ (1)

where, the window width is $(2k+1)$ is one of the major tuning parameter for this filter. x_n and y_n are the nth sample of the input and output sequences. The filter is fast in terms of computational time and not really difficult to implement.

2.2 Trend analysis

Trend analysis, also known as temporal reasoning, is a very important tool for diagnostics and decision support in complex industrial processes. Typically humans are very good at detecting patterns and trends in historial data by visual inspection. This is the backbone of any manual supervision and monitoring strategy of a industrial plant. However, detecting pattern by a automated algorithm is a difficult
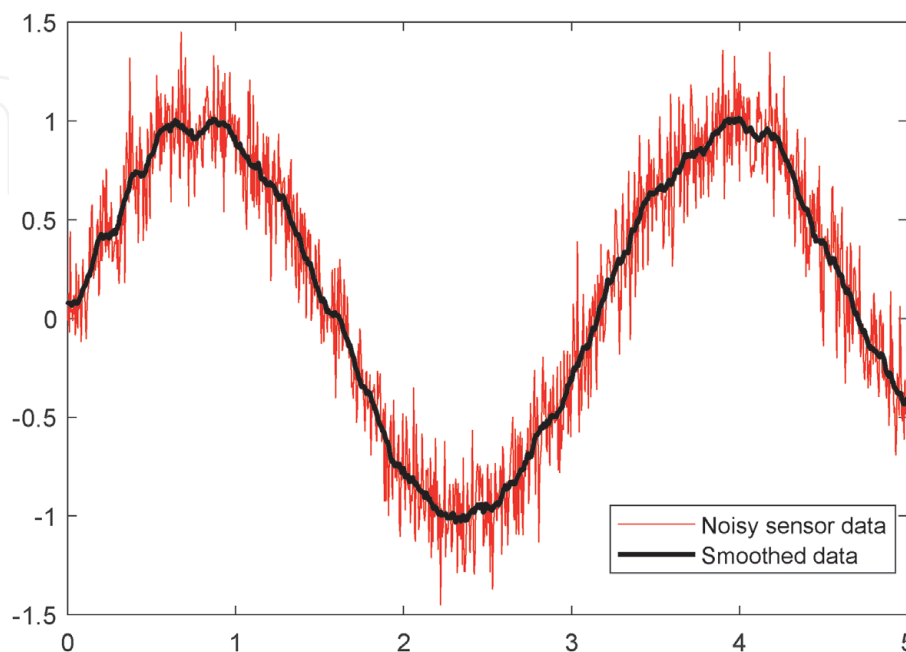
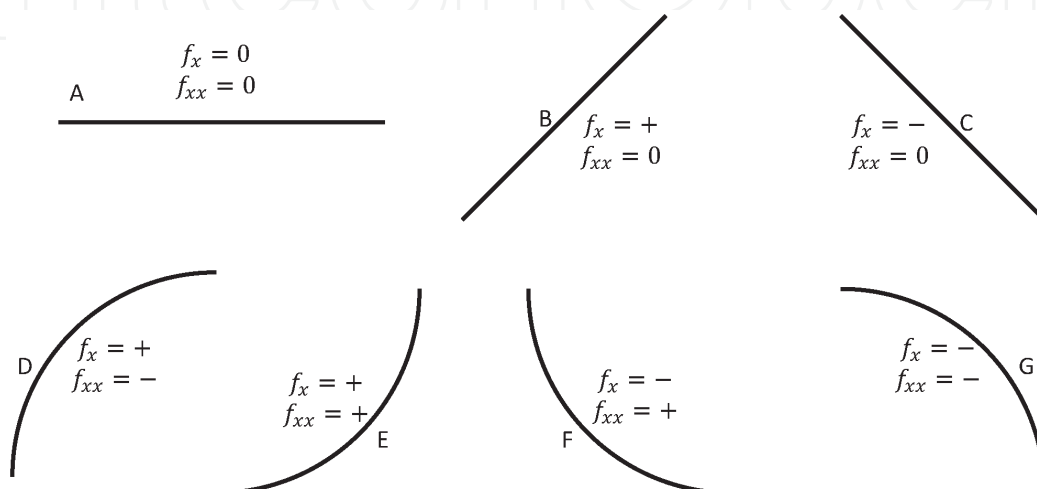**Figure 6.**
Example of noise reduction.

problem. Generally, trends are difficult to quantify due to the non-deterministic artifacts and background noise that typically presents in measurements. With the fast evolution advanced data analytics, it is possible to identify meaningful trends in time-series data that can be used in automated process monitoring, diagnostics and decision support. Numerous methods exist for performing trend analysis, ranging from the relatively simple methods such as linear regression to more complex methods such as Mann-Kendall and Spearman's rho tests to identify nonlinear trends in time-series data.

In this work, the aim of trend analysis is to extract useful trends from the historical process data so that it can be used as a prior knowledge to the decision support system. Moreover, visualizing automated trend information to the operators can improve their reaction time to any unwanted process drifts and abnormalities. The trend extraction methods can be either qualitative or quantitative in nature. Qualitative methods has gained upper hand over quantitative methods on extracting high-level knowledge from the process data [11]. Hence, Qualitative methods are better suited as the input needed by the decision support system. As the name suggest, qualitative trend analysis attempts to provide qualitative patterns from the historical data by fetching the underlying short and long term trends.

The most common way of representing qualitative trends in data is the use of seven primitives (**Figure 7**) with constant signs of first and second derivatives, originally developed by Janusz and Venkatasubramanian [12]. However, Charbonnier and Portet [4] proposed a self-adaptive qualitative trend analysis method by utilizing the first three primitives: steady (A), increasing (B) and decreasing (C). The method is further developed and applied to many industrial applications [13, 14]. The method divides online process data into linear segments to extract underlying trends. Real-time self-adaptation of the tuning parameters are performed to detect the variations and artifacts presents in the data. An example of trend fitting by using self-adaptive qualitative trend analysis approach is shown in **Figure 8**.

2.3 Process models

Process models, also known as mathematical models or simply models, are abstractions of real processes or systems that are used to characterize behavior of the processes or systems, given that the inputs are known [15]. Typically, such models can be used for prediction, control, fault detection, etc. Depending on the the modeling approach, models can be widely classified as first-principle, empirical

Figure 7.
Most common primitives.
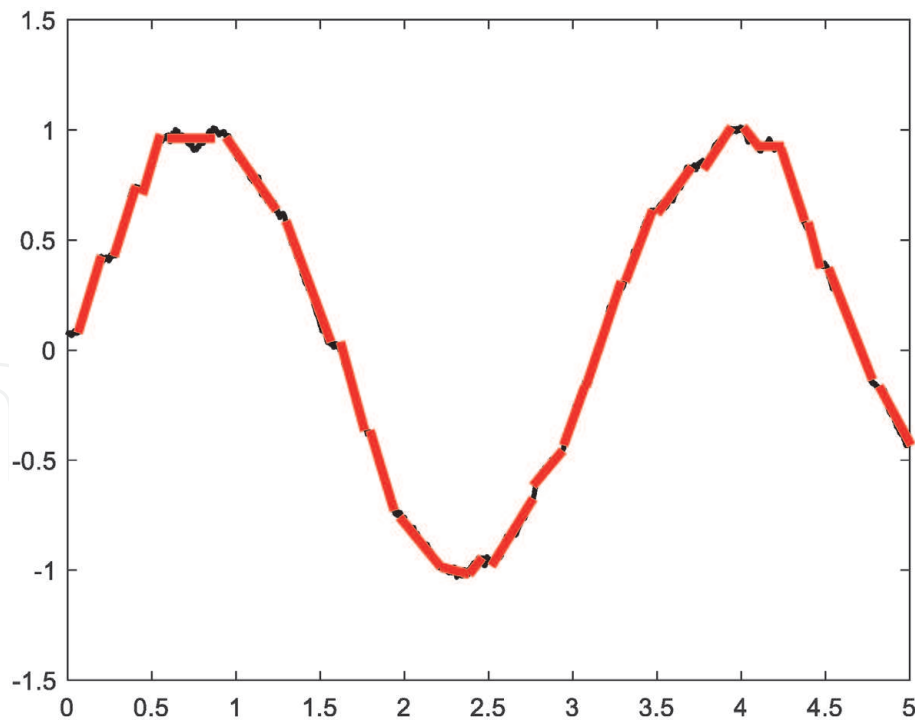
Figure 8.
Trend fitting example by self-adaptive approach.

and hybrid models. First-principle models, also known as White-box models, are based on mathematical equations that explain the physical, chemical or other basic principles. On the other hand, empirical models are based on data or observations that occurred in the past. These kind of models are also known as black–box models, as they relate inputs to outputs without revealing any knowledge of the internal working principles. Hybrid models are obtained by combining both first-principle and and empirical approaches. Process models can be further categorized into steady-state and dynamic models. A steady state-model is based on the assumption that the system is in equilibrium, and is thus time-invariant. This type of model is useful for system design but not for control applications. On the other hand, a dynamic model accounts for the time-dependent changes in a system and can therefore capture the transient behavior of the system. At the end the selection of the modeling approach and model types entirely depends on the purpose of the models. In this work, process models are used for output prediction, control and diagnostics purposes. Both first-principle and and empirical models are considered in order to take advantage of the benefits and avoid drawbacks associated with them.

The complexity of process models can vary widely, from simple conceptual models or linear models to high-fidelity computational fluid dynamics (CFD) models, depending on the purpose of the modeling work. Added model complexity almost always comes with a cost of high computational time that may impede the online application. There is no common modeling approach that fits the needs of all applications. Rather, the modeling approach for each application needs to be selected on the basis of the relevant purpose.

Typically, all theoretical process models are based on general conservation principles i.e. mass, energy and momentum balances, chemical kinetics, physical phenomenon such as friction, diffusion, compaction, and/or component specification. Most of the modeling work start with the assumption that some property is conserved within the system boundary. The general conservation principle can be formulated as Eq. (2).

$$\begin{array}{c}\text{Rate of} \\ \text{accumulation}\end{array} = \begin{array}{c}\text{Rate of} \\ \text{inflow}\end{array} + \begin{array}{c}\text{Rate of} \\ \text{generation}\end{array} - \begin{array}{c}\text{Rate of} \\ \text{outflow}\end{array} - \begin{array}{c}\text{Rate of} \\ \text{consumption,}\end{array} \tag{2}$$

Assuming the physical property under consideration is $X(t)$ where t is the independent variable for representing time. If the rate of change in inflows and outflows are denoted by $\dot{x}_{in}$ and $\dot{x}_{out}$; and rate of change in generation and consumption are denoted by $\dot{g}(t)$ and $\dot{c}(t)$ within the system boundary shown in **Figure 9**, a general balance law can be written in following form as in Eq. (3).

$$\frac{dX(t)}{dt} = \dot{x}_{in} + \dot{g}(t) - \dot{x}_{out} - \dot{c}(t), \tag{3}$$

This general balance law can be adapted for all three fundamental quantities: mass, energy and momentum, in order to model different industrial processes.

Reaction kinetics modeling another important aspect of process model development. For simplification let us consider a chemical reaction (Eq. (4)), where product C is formed by the reaction between reactants A and B.

$$A + B \rightarrow C, \tag{4}$$

Typically, the rate of reaction for a chemical reaction depends on principal quantities like temperature, pressure, and composition. For the sake of simplicity, let us assume that the effect of pressure is negligible in this case. Hence the rate of reaction r_c can be expressed as Eq. (5).

$$r_c = k * C_A{}^{\alpha} C_B{}^{\beta}, \tag{5}$$

where C_A and C_B are the concentration of reactants A and B, and k is the reaction rate constant. α and β are the exponents of concentration corresponding to each reactant. The rate constant k and the exponents α and β must be determined experimentally by monitoring how the rate of a reaction changes as the concentrations of the reactants are changed. The reaction rate constant k is temperature dependent and generally expressed according to the Arrhenius equation (Eq. (6)),

$$k = A_i * e^{-\frac{E}{RT}}, \tag{6}$$
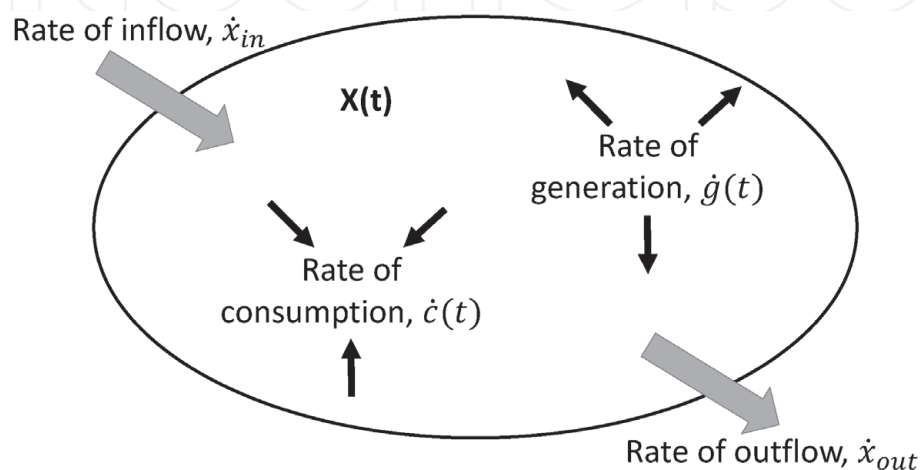
Figure 9.
*An example of a system boundary within which physical properties are considered to be conserved.*

Where, A_i, E, R and T pre-exponential factor, activation energy, universal gas constant and temperature of the reaction, respectively.

Depending modeling purpose the process models can also include more complex physical phenomenon such as diffusion, friction, compaction, porosity, velocity etc. A first-principle model usually consists of the three types of equations i.e. algebraic equations (AEs), ordinary differential equations (ODEs), and partial differential equations (PDEs).

Typically, dynamical systems are described by differential equations. Often a set of AEs are solved to find numerical solution of a set of ODEs. Generally, PDEs are used to describe processes with distributed parameters [16]. Partial derivatives with respect to both time and space resulted in models that computationally expensive to solve. Often lumped approximation is considered by assuming infinitesimally small continuous stirred tank reactors (CSTRs) in series. By assuming ideal mixing, it is possible to avoid changes of parameters in space inside a infinitesimal CSTR. Consequently, it is possible to model the process by using differential-algebraic equations (DAEs). DAEs are commonly solved by using various numerical methods. Many dynamic system modeling tools use their own solvers for this purpose. For example, One of the popular solver used by OpenModelica and Dymola is DASSL. The basic principle of DASSL is not unique, it replaces the derivative part with a difference approximation and solve the resulting system of equations with a Newton method [17]. However, great care in parameter initialization is necessary to ensure numerical convergence or fast convergence.

2.4 Model predictive control

Model predictive control (MPC) refers to a range of control algorithms for feedback and feed-forward control based on the receding horizon philosophy, where a set of optimal control moves are calculated according to the prediction of future behavior of the plant based on a process model. Using a process model, the MPC optimizer is able to estimate the consequence of past inputs on future outputs. As presented in **Figure 10**, at every control step, the MPC attempts to optimize future behavior of the plant by evaluating future sequential control moves over the prediction horizon. The controller then only executes the first step of the previously evaluated optimal control moves. The entire process is repeated again before the next control move.
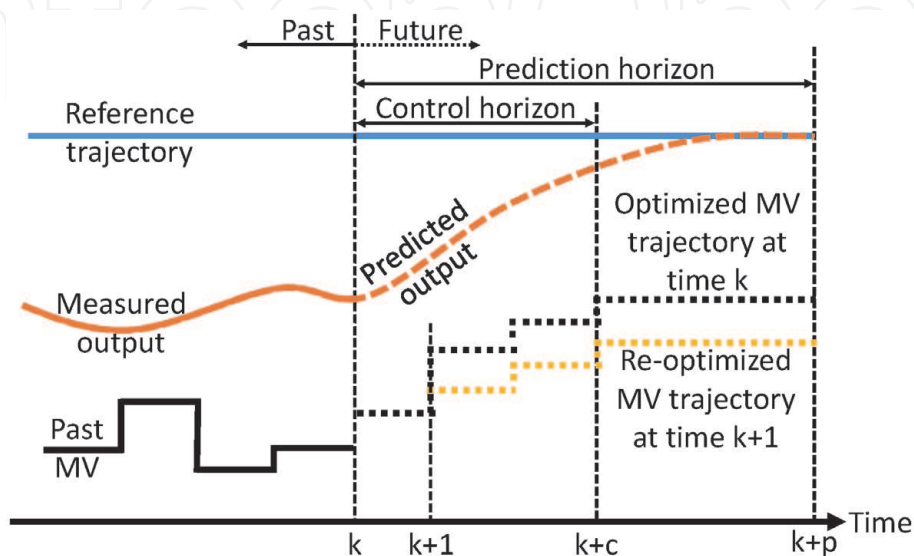
Figure 10.
Schematic representation of model predictive control. MV: Manipulated variable.

MPC provides superior performance particularly for processes that have multivariate interactions between inputs and outputs, which is a common traits of complex industrial systems. We argue that for highly complex processes MPC alone might not be the solution; particularly for processes where feed-stock properties varies unpredictably due to natural variation. In such cases, feed-forwarding the feed-stock variation to MPC will provide tighter control of the process. The scheme for a feed-forward MPC concept is depicted in **Figure 11**, where the feed-stock properties are feed-forwarded along with plant measurement to the MPC. A process model utilizes these information to make better prediction about the future outputs. The MPC optimizer computes the optimal control moves by solving a constrained finite-horizon optimization problem in which the cost functions make use of model predictions. The operational constraints are incorporated in the optimizer to ensure compliance. Additionally, the MPC also uses feedback to compensate for inaccuracies in the model and ensure convergence.

In reality, the cost function is a mathematical expression that is either minimized or maximized to find a best solution among all possible feasible solutions. Here, the cost function is expressed to find a sequence of incremental manipulated variables (MVs) over a control horizon of c samples, as presented in Eq. (7). The cost function minimizes a weighted sum of future squared errors of the outputs $y(k + i)$ and a weighted sum of increments in the sequence of MVs $\Delta u(k + i)$, while limits for Mvs and limits for predicted process variables are considered as a form of constraints in Eq. (8).

$$f(k) = \sum_{i=1}^{p} \|\Gamma e(k + i)\|_2^2 + \sum_{i=0}^{c-1} \|\wedge \Delta u(k + i)\|_2^2, \tag{7}$$

subject to constraints,

$$
\begin{aligned}
y_{min} &< y(k + i) < y_{max} & \forall i\, [1, p] \\
u_{min} &< u(k + i) < u_{max} & \forall i\, [0, c - 1] \\
\Delta u_{min} &< \Delta u(k + i) < \Delta u_{max} & \forall i\, [0, c - 1]
\end{aligned}
\tag{8}
$$

In this minimization, the future errors $e(k)$ are calculated over a prediction horizon of p samples according to Eq. (9),
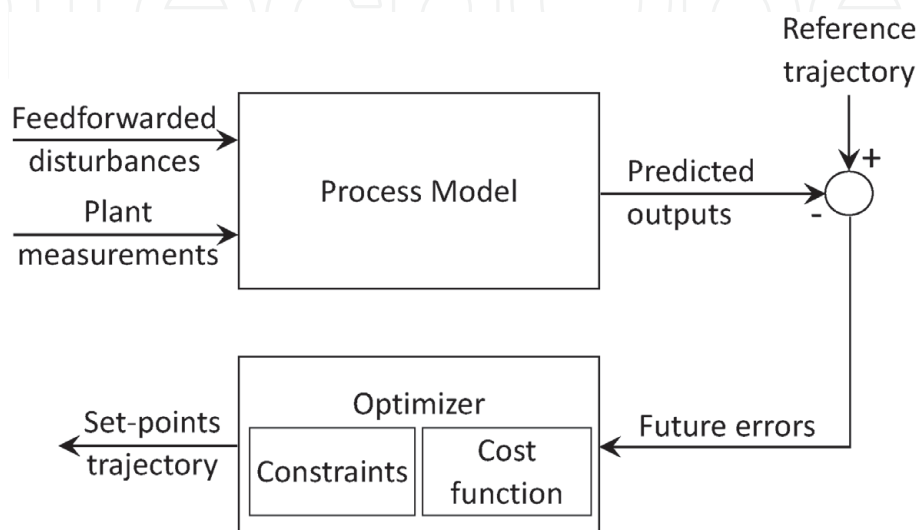
Figure 11.
*Scheme for feed-forward MPC.*

$$e(k) = s(k) - y(k), \tag{9}$$

where, $s(k)$ denotes the reference set-point trajectories. For outputs prediction a dynamic model can be used, or if an observer based formulation is utilized then a reduced order state-space model identified from the dynamic model, or linearised model from the step tests of a real plant can also be used.

2.5 Anomaly detection

Anomalies are the unusual, unexpected, abnormal patterns in a signal or a process variable. The term anomaly comes from the Greek word "anomolia" that means uneven or irregular. So how anomalies differ from faults? Faults are unexpected malfunctions in one or more components of a process that are not a failure or breakdown. However, faults may result in failures or catastrophic breakdowns if not resolved in time. On the other hand, anomalies only tell us there might be something abnormal with the system or a signal but it not necessarily means there is a fault in the system. Anomalies can occur due to many reasons other than faults. Maybe it is hard to accept but real systems are continually anomalous in many ways. Interestingly, anomalies can be positive or negative in nature depending on the context and interpretation [18]. Due to its nature, anomaly detection creates significant noise. However, detection of such abnormal conditions in the process can assists the operators on decision making so that they can react in time to avoid or correct the situations associated with them. Here, for the decision support system, anomaly detection is an additional source of information that will assist in robust decision making. We will also take the opportunity to distinguish between outlier detection and anomaly detection functions as we use both of these techniques in our framework. In outlier detection, we detect and remove or replace data that are either missing or illegitimate (e.g. a negative flow-rate) or very far away from rest of the data. In anomaly detection, we detect abnormal pattern in data and forward this information to the decision support system.

According to the general failure mode curve (**Figure 12**), a new machine runs with good health condition for some period of time. Then it reaches a point H where
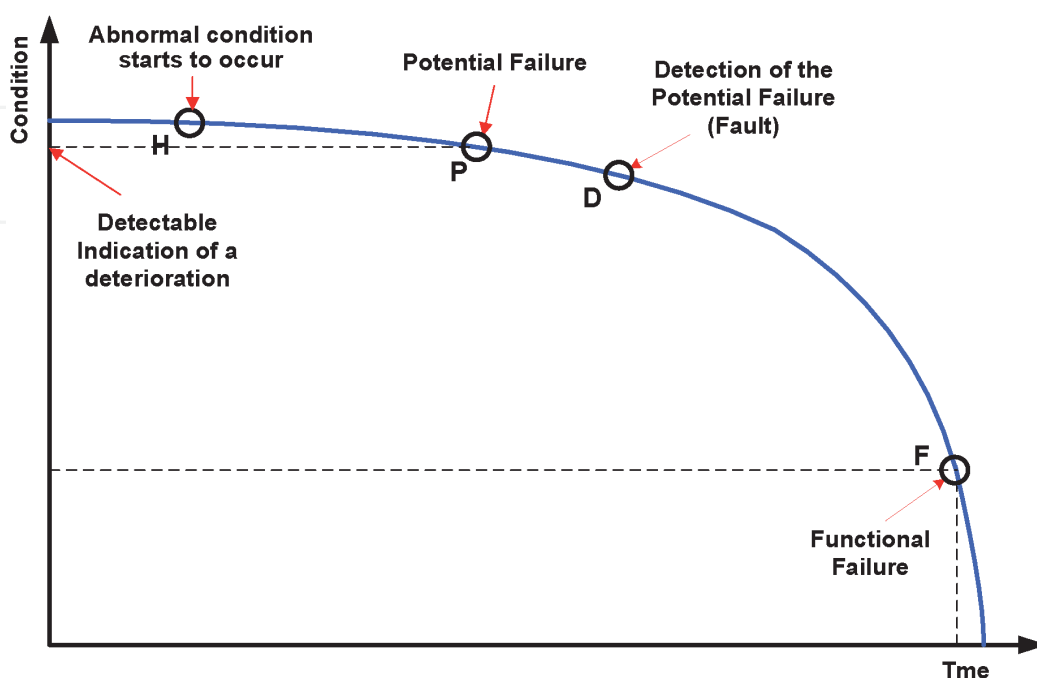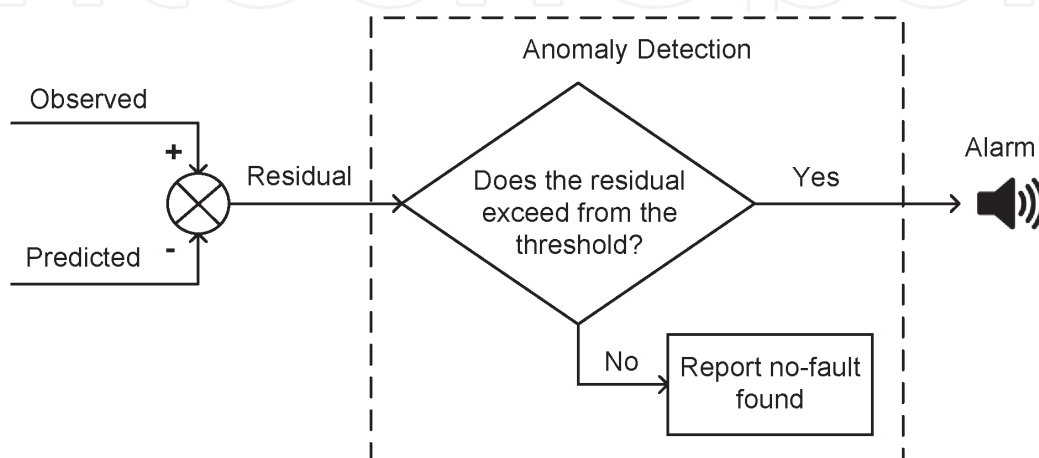

Figure 12.
Equipment failure mode diagram (adapted from [19]).

degradation starts to occur due to some damage-causing conditions. Point P represents the time where potential failure is recognized. The degradation progresses and then reaches a point where it can be detected. In general, the abnormal condition between P and F falls within the detectable range. The range between P and D refers to anomaly whereas between D and F refers to a fault [19]. Anomaly can also be a discrete event causing a rapid shift in measurement changes [20]. The goal of anomaly detection is, therefore, to detect the potential failure as early as possible.

Anomaly detection is extensively studied within many different application areas including credit card fraud detection, finance, cyber-intrusion, network monitoring, and many industrial plant monitoring [21, 22]. The simplest form of industrial anomaly detection technique can be as simple as logging an alarm if a sensor reading drifts away from a predefined upper and lower boundary. However, there are quite many anomaly detection techniques explored by researchers; which can be broadly categorized in three groups: (a) statistical techniques i.e. principal component analysis (PCA), histogram, Gaussian mixture models, Gaussian Kernels, etc., (b) cognitive techniques i.e. expert systems, finite state machine, etc., and (c) machine learning techniques i.e. clustering, classification, etc.

Anomaly detection is an important step in the process of fault diagnostics, and can be performed using measurement deviations or residuals as illustrated in **Figure 13**. A threshold-based detection or a binary logic can be applied. According to the threshold-based anomaly detection, the residuals should be very close to zero or lie within the threshold when the system is running in a normal operating condition and at least one residual should deviate noticeably from zero when an anomaly occurs. As a threshold, a Gaussian distribution of the residuals is often assumed in order to take into account variations due to measurement uncertainties. In the case of the binary logic, the residual is considered as a signal which is zero when the system is functioning properly and different to zero when some abnormal behavior is observed.

There are a variety of methods available for anomaly detection starting from the conventional model-based or statistical approaches to the more sophisticated machine learning techniques. Model-based methods rely on system models combining the theoretical knowledge with the test or actual performance data. When an abnormal condition (or a discreet fault event) occurs somewhere in the system, it produces deviations in measurements from their expected reference values. An accurate system modeling followed by a robust residual generation and proper threshold selection is critical. Machine learning techniques usually treat the anomaly detection task as a pattern recognition problem. The algorithm tries to learn a decision boundary from

Figure 13.
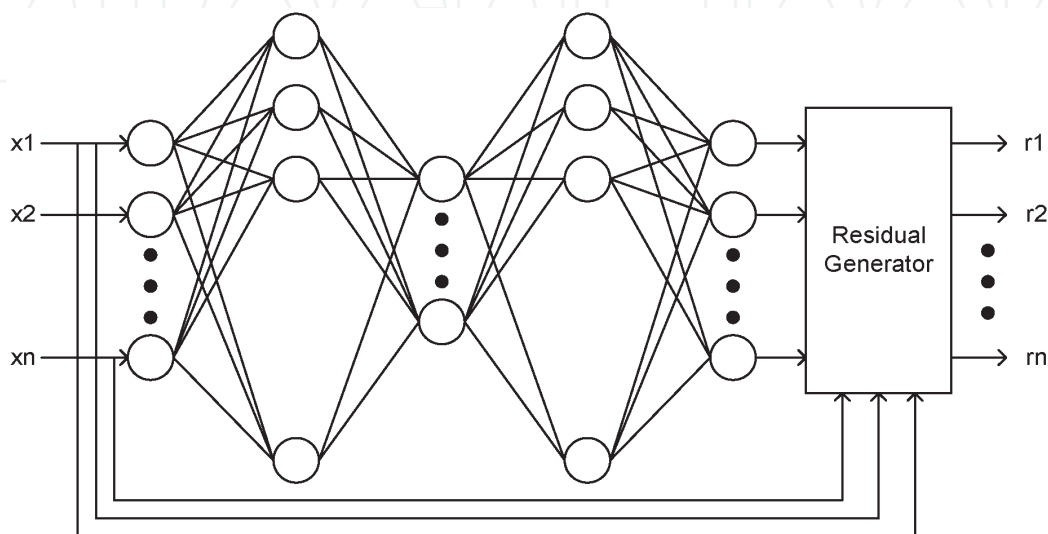*Anomaly detection schematics.*

the training data (i.e. the normal data). The detection accuracy can be evaluated using the standard detection decision matrix as presented in **Table 1**.

In machine learning algorithms anomalies can be detected in a supervised or unsupervised way. In the former case, labeled data is used for training. The labels can be binary, e.g. yes/no, one/zero, normal/abnormal, and fault/no-fault. ANN, support vector machine (SVM), and K nearest neighbor (KNN) are examples of the widely used supervised classification algorithms. For the unsupervised case, the normal and abnormal classes are distinguished based on their similarity using distance or density functions. Hierarchical clustering (HC), self-organizing map (SOM), K-means and K-medoids are some of the common unsupervised clustering algorithms.

In ANNs, a fault detection task is considered as pattern recognition. During training sample patterns of the two classes are feed into the network and the network tries to recognize the patterns based on their corresponding output labels. Among ANNs, an autoassociative neural network (AANN) is more suitable for an anomaly detection [23]. First, the model is trained on a normal data as input and output (**Figure 14**). For the normal input data, the difference between the model output and the target output will be close to zero, while for abnormal input patterns, at least one of the output residuals will deviate noticeably from zero.

| Actual	Predicted				
	Abnormal	**Normal**	**Total**		
Abnormal	True abnormal	False normal	K1 + K2	Detection rate	Missed detection rate
	K1	K2		K1/(K1 + K2)	K2/(K1 + K2)
Normal	False abnormal	True normal	K3 + K4	True normal rate	False alarm rate
	K3	K4		K4/(k3 + K3)	K3/(K3 + K4)
Total	K1 + K3	K2 + K4	K1 + K2 + K3 + K4	Detection accuracy	
				(K1 + K4)/ (K1 + K2 + K3 + K4)	

Table 1.
*Detection decision matrix.*

Figure 14.
Architecture of an AANN for anomaly detection.

According to KNN algorithms, anomalies are data points located farthest away from the normal data points or in low-density regions if weighted distances are considered (see **Figure 15**). After estimating all distance values, they need to be sorted in descending order. Anomalies are data points with the largest distance values. Then, the test data points that fall in the top n% distance range are considered as anomalies, where n is user defined value. The Euclidean function is the most convenient distance function in KNN.

A support vector machine is another type of supervised learning classifier. It is a binary classifier in its nature that separates two different classes by maximizing the margin between them. If one of the classes to be distinguished is taken as positive the rest of the class will be considered as negative. The classifier will, therefore, learn a boundary to separate the positive and negative classes as illustrated in **Figure 16**. The purpose of the support vector machine is to maximize the separation distance (margin) between the two classes. The type of SVM used for anomaly detection is called a one-class SVM. In this case, the model is trained only on the normal data class, and anything deviated from the normal class is considered an anomaly. The one-class SMV maps training data patterns into a high-dimensional
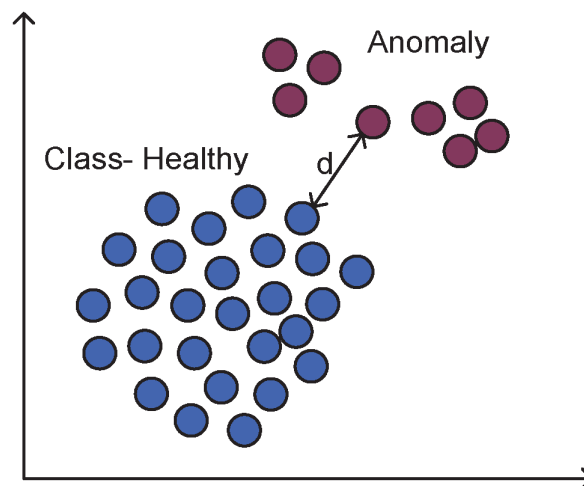
Figure 15.
Anomaly detection using a KNN.
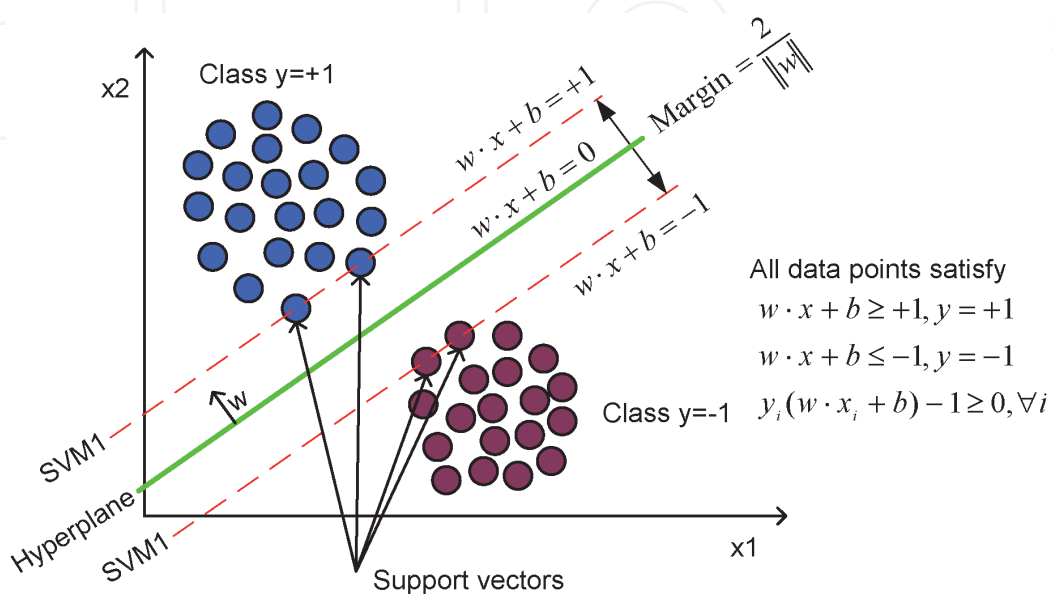
Figure 16.
The SVM classifier for linearly separable classes.

feature space using the kernel function and finds the maximum margin that separates the training sample and the origin. **Figure 17** shows a linear one-class SVM for data points in a 2-dimensional space.

The function to be minimized in order to maximize the margin between the origin and the training class is

$$\min_{w,\xi,\lambda} \ \frac{1}{2}\|w\| + \frac{1}{vm}\sum_{i}^{m}\xi_i - \lambda \quad \text{s.t.} \quad (w,\psi(x)) \geq \lambda - \xi_i, \xi_i \geq 0 \quad and \quad 0 < v < 1,$$

(10)

The decision function is given as

$$f(x) = sgn\left(<w,\Phi(x)> - \lambda\right),$$

(11)

Applying Lagrange multiplication yields the following quadratic programming to be optimized

$$\min \ \frac{1}{2}\sum_{i,j}\gamma_i\gamma_j k(X_i.X_j) \quad \text{s.t.} \quad 0 \leq \gamma_i \leq \frac{1}{vm} \quad and \quad \sum_{i}^{m}\gamma_i = 1,$$

(12)

where γ is the Lagrange multipliers, k is the kernel function used to project the input feature into the feature space, λ is an offset parameterizing a hyper-plane in the feature space, and m is the number of training data points. There are different types of kernel functions for instance, linear kernel, polynomial kernel, radial basis function (RBF) kernel, and Sigmoid kernel.

2.6 Fault diagnostics

After detecting an anomalous condition, fault diagnostics, also known as process diagnostics, aims to determine and provide specific information about the possible cause. Often this process can be also quite independent from the anomaly detection
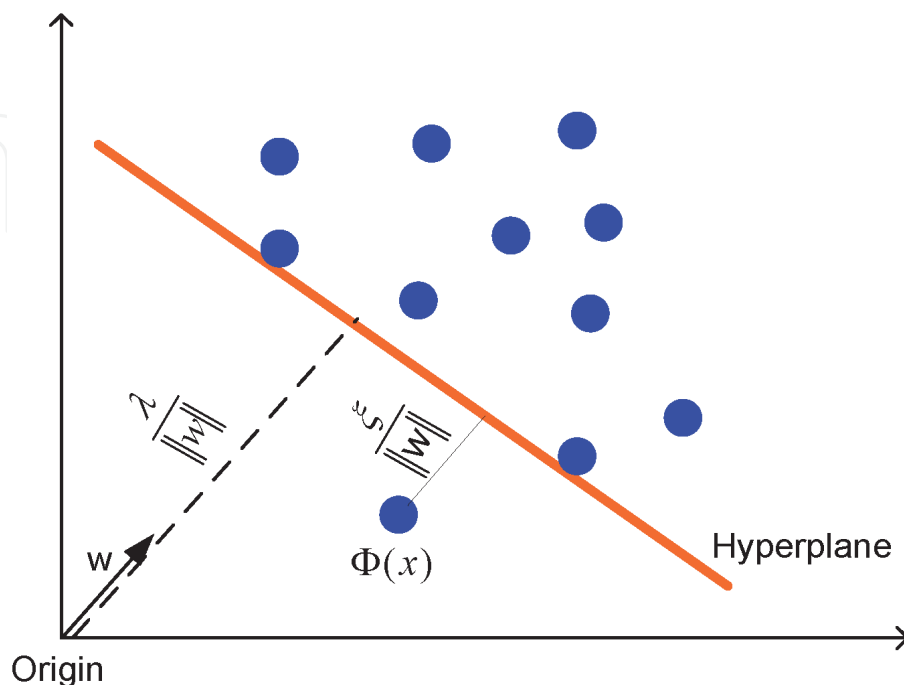
Figure 17.
One-class SVM.

layer. Typically, process diagnostics consists of three steps known as fault detection, isolation, and identification. Fault detection and isolation (FDI) are often performed simultaneously by use of physical models or data-driven techniques, as in the case of anomaly detection. Fault detection is the step to determine if a fault is present or under development in the process and the time of fault occurrence. Fault isolation refers to the technique of pinpointing the location of the faulty component(s) of a process, such as devices, sensors, actuators, controllers etc.

Typically, FDI methods are widely classified in three categories: model-based methods (typically first-principles, state-space or input–output models), model-free (also known as data-driven) methods, and knowledge-based (or rule-based) methods. All these methods have their own advantages and disadvantages. Hence, in the realm of process diagnostics, there is no silver bullet to address every single case. For applications where the process is difficult or expensive to model, or sufficient information is not available to model the effects of all possible anomalies and faults accurately, data-driven and machine learning methods have been developed in the years. Similarly to the techniques used for anomaly detection, classification techniques such as ANN, KNN and SVM are often used to assign the measured data points to the cluster indicating the faulty component. However, simultaneous faults or malfunctioning in more than one component require more complex methods. All the methods have various degrees of sensitivity to measurement noise.

On the other hand, fault identification refers to the way of estimating the severity or magnitude of the fault, and providing information on whether the process can continue to operate as usual or if a corrective action (in extreme case, shut-down) needs to occur. Typically, the extent of deviation in measured parameters can give an indication on the severity of the fault. However, simultaneous faults in different components may have opposite effect on measured parameters, hiding the real problem magnitude and rendering this step quite challenging.

A combination of model-based and data-driven approaches for fault isolation and identification is often preferred when an accurate numerical model of the process is available. A common approach is to include health indicator factors, or state variables, in the model. Health indicator variables can represent e.g. a fouling coefficient in a heat exchanger or a flow capacity deviation in a pump, compressor, or turbine. Such variables can be varied when simulating the process until the model outputs match the observed measurements from the real system. Various optimization techniques such as genetic algorithm (GA) have been used for this purpose. This method can often perform isolation and identification together, when the health indicator factors are allowed to take multiple values. One drawback commonly experienced is the so-called smearing effect, mostly induced by noise and model uncertainties, where the effect of anomalous measurements tends to be "spread" over multiple health factors even when only one single component is actually faulty. To overcome this, preprocessing of the data to reduce noise is usually necessary; downstream processing of the obtained health indicators through machine learning techniques is also a solution to improve the isolation and identification accuracy.

2.7 Advanced sensors

The learning system is designed to work with the data that is collected in the database. In order to best utilize this, it is important to understand the inherent properties and qualities of the data gathered about the process. Data is gathered from multiple sensors located in different parts of the process. Sensors are devices that provide output signals based on a certain input that represents a physical

quantity. These devices can be more or less complex, ranging from straightforward ones that measure pressure and temperature to more complex ones that determine other physical and chemical properties. The most measured parameters in the process industry are temperature, followed by pressure and flow rate. However, this does not mean that these parameters can be measured accurately.

Sensors are based on different principles to provide useful information about the system. Most physical properties of interest cannot be measured directly but are rather obtained by utilizing different principles, converting a property that can be easily measured to the one of interest. Temperature is often measured with a thermocouple, which utilizes the thermoelectric effect where temperature differences are converted to electric voltage. Two dissimilar metal wires are connected at one end in an electric junction. Once the temperature changes at the junction, it creates a voltage that can be measured with a voltmeter and is a function of temperature. This is the case for the majority of sensors used in the process industry, requiring some sort of model to convert the measured parameter to useful output.

Measurements of a process parameter carry with them a certain uncertainty. This can arise from many sources and will propagate to the final result. The more the sources of uncertainty and the more complex the process of getting to that result, the higher the final uncertainty in the measured data. This difference in the measured value from the actual one can be either random or systematic. Random differences contribute to random signal noise, whereas systematic differences can be due to bias or deterioration of the sensor. An unsteady process, where the values of the parameters fluctuate, will make it even harder to obtain accurate measurements.

For the development of a learning system, measurement data from the process can be used to monitor the operation of the different modules. This can provide information on the performance of the different components. Another very useful piece of information, particularly for the process industry, is information on the properties of the feedstock that is coming into the process. This can provide a feed-forward signal, enable the prediction of the properties of the final product, and allow the optimal control of the process. This can be done with a more advanced sensor, which in essence requires a more complex model to convert the measurement to the property of interest. Such sensors are often referred to as soft sensors. As long as the sensor and the property of interest are in the same location in the process, the model is part of the sensor, regardless of its complexity. If the property of interest is in a different part of the process than the one where the sensor is installed, the model behind the sensor becomes a model of the process rather than a conversion of the sensor input to useful output. These advanced sensors are subject to uncertainty and noise in the same way as the simpler ones. However, uncertainty and noise in the measurement can increase when there are more components and models in a measurement chain, and this affects how the data can be used.

With regards to the measurement of feedstock properties, a particularly promising technology is near infrared (NIR) spectroscopy. This technique is based on the difference in absorbance of light in the near infrared field by different chemical bonds in the molecules that are illuminated. This can in turn provide detailed information about the chemical properties of the material that is measured. The measurement head itself will provide a spectrum of absorbance in a range of wavelengths and this information can be calibrated against the desired physical properties of the material that are of interest for the specific application. This is typically done at first in a laboratory environment, and the models obtained from the laboratory experiments are then transferred or adapted to the real environment. NIR has been shown to be very capable of predicting key properties of the incoming feed

for a range of different processes [24–26] in a much faster way than time consuming lab analyses and as such can form the base of an advanced sensor in the process industry.

An example of a lab setup for the analysis of fuel samples with a Fourier transform NIR spectrometer is shown in **Figure 18** for refuse-derived fuel (left) and woodchips (right). NIR spectroscopy can be used for both solid and liquid fuels, and a spectrum of a hydrocarbon mixture is shown in **Figure 19**. The spectra obtained from the NIR instrument are matched to the property of interest for every sample analyzed and a calibration model is built using a statistical analysis of the data (typically referred to as chemometrics for spectroscopy applications). This results in a calibration curve like the one shown in the right hand side of **Figure 19** where the quantity of interest predicted by the model from the spectrum of the sample is compared to the quantity of interest measured in the lab. The 45° line in the figure represents a perfectly accurate prediction (the predicted value is the same as the measured one), but small deviations from the measurements always occur and the accuracy of the model is depicted by the width of the area between the dashed lines.

In order for the NIR-based soft sensor to be used in a real process environment, the head needs optical access to the feedstock. An example of an installation of a NIR sensor in a pulp and paper mill is shown graphically in **Figure 20**. In this case the optical access is provided through an observation hole. The NIR spectra obtained by the instrument are converted to the desired property through a model,

Figure 18.
Lab setup for the analysis of fuel samples with a Fourier transform NIR spectrometer for refuse-derived fuel (left) and woodchips (right).
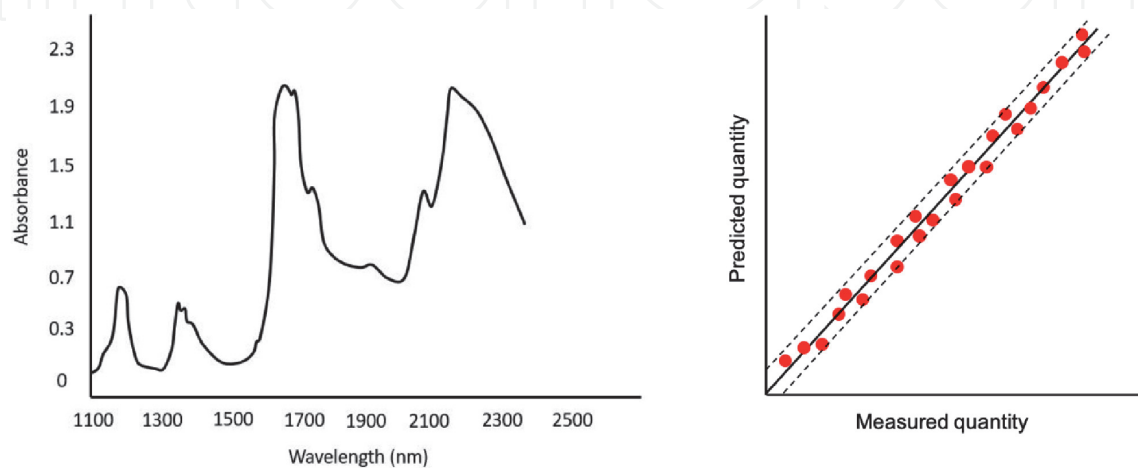
Figure 19.
NIR spectrum of hydrocarbon mixture (left) and soft sensor model calibration (right).
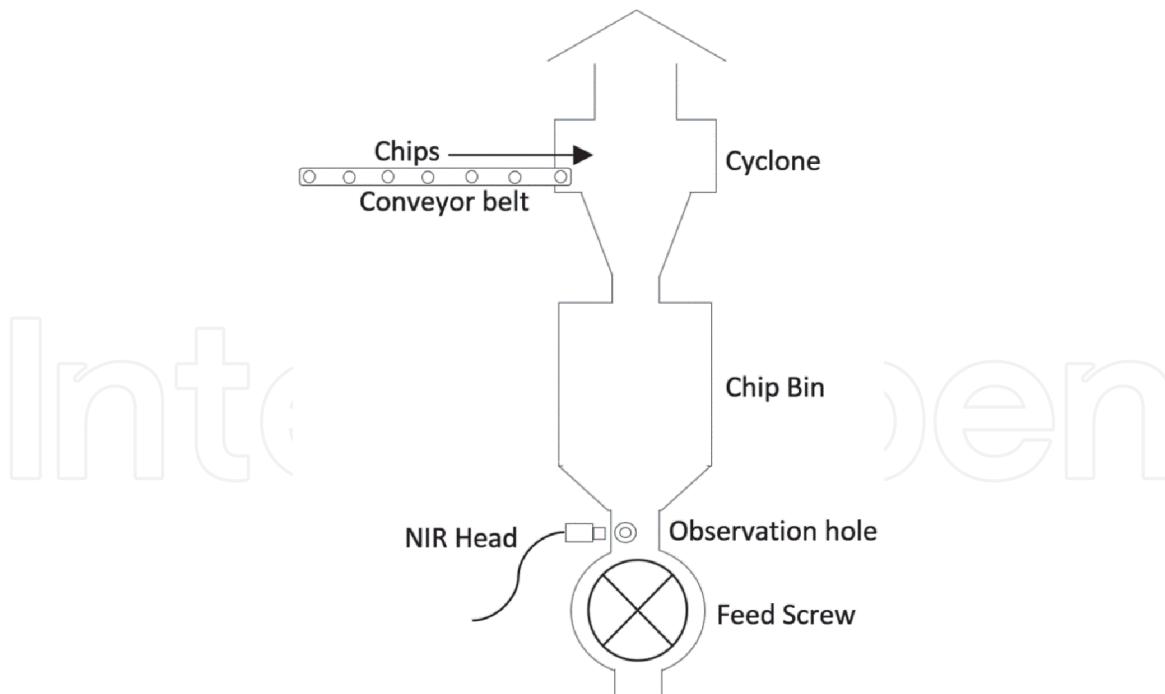
Figure 20.
Schematic of the NIR soft-sensor location in a pulp digester.

with the entire setup of the measurement head, the analyzer and the model behind it constituting the soft sensor. The information from the sensor can be delivered to the database of the factory and from there it can be fetched and used as a feed-forward input signal for the control of the process.

The main advantage of using a NIR sensor is that it can provide fast and non-intrusive measurements of the properties of the feedstock material, which can then be used to optimize the operation of the process itself. Variations in the properties of the feedstock will inherently affect the process and the optimal operating conditions of the downstream components will change depending on these properties. Real-time information on the variation of the key properties of the feedstock can therefore be very useful when combined with model predictive control in order to determine the optimal operating conditions in real-time, with the information from the sensor used as a feed-forward signal for the control of the process.

2.8 Decision support system

A decision support system (DSS) is a computer-based program that supports decision-making activities, for example at operation, planning or management levels. Through the analysis of large amounts of data, a DSS provides decisions for uncertain, unstructured, or rapidly changing problems, which either complement or replace human reasoning. The system can be either fully automated, fully dependent on human actions, or a hybrid. However, the hybrid approach is widely acceptable where the DSS incorporate a human–computer interaction and the cyber part usually provides a range of information that operators or managers use to decide on an action [27]. A DSS can for example have access to a database of historical events and corresponding decisions, and retrieve cases similar to a current event to suggest a possible action. If we linked it back to the automation pyramid, typically a DSS will directly interact with the supervisory level and influence the decision making in planning and management level. Within the learning system architecture, the DSS is used to support decision making at all three levels. Based on the outcome from individual sub-components of the learning system, the DSS will

assists the operators by providing information about the health status of the process equipment. The system can also suggest possible actions under a particular process fault. This can also trigger series of actions that will assist decision making at planning and management level. Due to a fault severity if a equipment requires maintenance, the DSS will assist the maintenance planner by providing RUL of the affected equipment. Additionally, if a process equipment becomes unavailable the production planning need to be adjusted accordingly. This will effect the decision making in ERP level since both production material and spare-parts inventory need to be adjusted accordingly.

According to literature, DSS can be model-driven, data-driven, knowledge-driven, document-driven or communication driven [28]. A model-driven DSS employs statistical, financial, mathematical, analytical, simulation or optimization models for decision support. In model-based DSS, possible scenarios are simulated with the aid of models to take optimal decisions; for example, the optimal maintenance interval can be calculated with the aim of minimizing the total costs. Developing a model-driven DSS is a complex, time consuming and expensive process that requires a considerable level of expertise. A data-driven DSS requires access to and manipulation of time series of internal and external data. It is the most common of the five types of DSS. The success of a data-driven DSS always depends on the access to accurate, well-structured and organized data. A typical knowledge-driven DSS contains a rule-based algorithm such as decision tree or similar [29]. The knowledge from the expert is stored in form or rules such as "if sensor A is faulty and control system is functioning, schedule maintenance in XX hours". Thus, automated decisions can be taken by analyzing massive amount of data and applying predefined rules. In this work, we will only focus on knowledge-driven DSS particularly highlighting an example of probabilistic approach.

More sophisticated algorithms aiming at simulating human reasoning in a probabilistic manner are built from Bayesian belief networks (BBN). Bayesian networks represent a culmination of Bayesian theory of probability, which can be summarized as in Eq. (6). The equation represents a casual statement of the kind, where X causes Y and Y takes the role of an observable effect of X. $P(Y)$ is called the prior probability, while $P(Y|X)$ is called the posterior probability. The factor that relates the two, $P(X|Y)/P(X)$, is called the likelihood ratio.

$$P\left(\frac{Y}{X}\right) = \frac{P\left(\frac{X}{Y}\right)}{P(X)} * P(X), \qquad (13)$$

A BBN is a probabilistic graphical model that represents factorization of joint probability distribution [12]. It provides a comprehensive way to handle uncertainty in mathematical computation, consequently widely used for representing uncertain knowledge. Bayesian probability differs from classical probability by the fact that classical probability does not put any weightage to the evidence while Bayesian probability always comprises of a certain degree of belief in the evidence [13]. The most beneficial aspect of a BBN is that it can be constructed either by training with historical data, and with limited data set or even in the absence of data only by integrating expert knowledge. A BBN has two major parts: a qualitative or structural part, consisting of nodes and connections, and a quantitative part that is a set of conditional probability distributions. Typically, each node corresponds to a unique random variable (e.g. occurrence X), while each edge or connection corresponds to a conditional dependency. This qualitative structure is referred to as directed acyclic graph (DAG). The term "acyclic" refers to the fact that the direct connections are static causal probabilistic dependence and cycles are not allowed (e.g. if X causes Y, Y cannot cause X). Constructing a BBN involves building the

structural part of the BBN or DAG and specifying the conditional probabilities also known as parameters. A BBN can be constructed completely manually from expert knowledge, completely automatically from data, or through a combination of a manual and automatic technique, where partial knowledge about the structure or the parameters are learnt from the data.

For maintenance planning purposes, the DSS would combine and fuse together information coming e.g. from different diagnostics approaches, maintenance history, operator observations, etc.

2.9 Applicability of the framework for fleet level monitoring

The presented framework can be applied both to single large units (ie a complex industrial plant) and to multiple smaller units, in a fleet management approach. The requirements for fleet management shape the framework in a multitude of ways.

In order to manage multiple assets at the same time, the level of detail of the simulation may be reduced. This may result in less complex models and different requirements of the level of control and management. This can further increase the modularity of the framework. Different levels of control and management may not be desired and approaches used may be less complex. In essence, a framework that focuses on fleet management will not focus on the optimization of a single system with a multitude of sensors at the first instance. This requires an approach that allows the removal or deactivation of functions as desired. This is in line with the development of a framework that can be generic and applicable to different cases, but further highlights the need for modularity.

The increase of the number of assets being monitored through the framework requires more models that simulate the operation of the assets. One option is to have a different model for each system, which can result in thousands of models being employed in the platform. However, the units of the fleet being managed are inherently very similar to each other; they are copies of each other with minor differences which arise from manufacturing uncertainties. The units will be operating at different conditions, which will affect the degradation of components and sensors in a unique way for each system. However, the different assets can be represented by a single model that simulates their performance and a different set of tuning parameters for each one. This can reduce the load in the framework platform.

The management of the fleet will result in large amounts of data being collected, a set of data for each asset in the fleet. Since the different assets belong to the same family, and their operation is similar, the data from the entire fleet is useful for the management of all assets independently and collectively. A system that learns from the operation of different assets can gain more knowledge than one that is focusing on a single asset. This creates more challenges for data management, but also provides more knowledge for possible faults, and can allow the prognosis of remaining useful life and other parameters of interest with greater confidence. From the framework perspective, this requires more instances for visualization of data from both the single unit and the fleet, and an analysis of the different trends.

3. Conclusions

Over the last few decades there has been an significant exploration of new techniques and tools to improve product quality and process efficiency of complex industrial processes. There is a need for a framework that will allow integration of different tools for optimal operation, control and diagnostics to enable robust

decision support. As a stepping stone, a generic learning system architecture has been developed that allows easy integration with existing supervisory system of industrial plants. The architecture enables inclusion of different functionalities as individual modules. The system can therefore be easily adapted according to the different requirements of different cases. The architecture is flexible enough to be implemented in a remote server with a web-based interface or run locally in a isolated server. As a final reflection, utilization of such a learning system in addition to the existing supervisory systems can only be justified by demonstrating quantifiable economic benefits. Only then will all stakeholders be on board for adoption of such a system. Another aspect that is often neglected is that the system users, i.e. plant operators, engineers and managers, need to be involved from the very beginning of the process from development to implementation of such a system.

Acknowledgements

Conflict of interest

The authors declare no conflict of interest.

Nomenclature

$X(t)$	Physical property under consideration
t	Independent variable for representing time
$\dot{x}_{in}$	Rate of change in inflow
$\dot{x}_{out}$	Rate of change in outflow
$\dot{g}(t)$	Rate of change in generation
$\dot{c}(t)$	Rate of change in consumption
A, B	Reactants
C	Product
r_c	Rate of reaction
α, β	Exponents of concentration
k	Reaction rate constant
A_i	Pre-exponential factor
E	Activation energy
R	Universal gas constant
T	Temperature of the reaction
c	Control horizon
$y(k+i)$	Future plant output

$\Delta u(k+i)$	Increments in manipulated variable
$e(k)$	Future errors
p	Prediction horizon
$s(k)$	Reference set-point trajectory
γ	Lagrange multiplier
k	Kernel function
λ	Offset parameterizing a hyper-plane in the feature space
X	Cause of Y
Y	Observable effect of X

Abbreviations

ISA	International society of automation
PLC	Programmable logic controller
DCS	Distributed control system
PID	Proportional–integral–derivative
SCADA	Supervisory control and data acquisition
HMI	Human-machine interface
MES	Manufacturing Execution System
ERP	Enterprise resource planning
CMMS	Computerized maintenance management system
RUL	Remaining useful life
HMI	Human–machine interface
CFD	Computational fluid dynamics
AE	Algebraic equation
ODE	Ordinary differential equation
PDE	Partial differential equation
CSTR	Continuous stirred tank reactor
MPC	Model predictive control
MV	Manipulated Variable
PCA	Principal component analysis
ANN	Artificial neural network
SVM	Support vector machine
KNN	K nearest neighbor
HC	Hierarchical clustering
SOM	Self-organizing map
AANN	Autoassociative neural network
RBF	Radial basis function
FDI	Fault detection, isolation and identification
NIR	Near infrared
DSS	Decision support system
BNN	Bayesian belief networks
DAG	Directed acyclic graph

Author details

Moksadur Rahman*, Amare Desalegn Fentaye, Valentina Zaccaria,
Ioanna Aslanidou, Erik Dahlquist and Konstantinos Kyprianidis
Malardalen University, Vasteras, Sweden

*Address all correspondence to: moksadur.rahman@mdh.se

IntechOpen

References

[1] Rahman M. Towards a learning system for process and energy industry: Enabling optimal control, diagnostics and decision support. Licentiate dissertation, Mälardalen University, Västerås, Sweden; 2019.

[2] Lee J, Davari H, Singh J, Pandhare V. Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. Manufacturing Letters. 2018; 18:20–23.

[3] Diez-Olivan A, Del Ser J, Galar D, Sierra B. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. Information Fusion. 2019; 50:92–111.

[4] Leitch R, Quek C. Architecture for integrated process supervision. In: IEEE Control Theory and Applications. vol. 139. IET; 1992. p. 317–327.

[5] Baldea M, Harjunkoski I. Integrated production scheduling and process control: A systematic review. Computers & Chemical Engineering. 2014;71:377–390.

[6] ANSI/ISA-95. Enterprise Control System Integration Part 3: Activity Models of Manufacturing Operations Management; 2005.

[7] Hummen R. What is TSN? A Look at Its Role in Future Ethernet Networks [Internet]; 2017. Available from: https://www.belden.com.

[8] Miskowicz M. Event-Based Control and Signal Processing. Boca Raton, FL, USA: CRC Press; 2018.

[9] Mehta BR, Reddy YJ. Chapter 19 - advanced process control systems. In: Industrial Process Automation Systems. Oxford: Butterworth-Heinemann; 2015. p. 547–557.

[10] García S, Luengo J, Herrera F. Data preprocessing in data mining. 1st ed. Cham, Switzerland: Springer International Publishing; 2015.

[11] Zhou B, Ye H. A study of polynomial fit-based methods for qualitative trend analysis. Journal of Process Control. 2016;37:21–33.

[12] Janusz ME, Venkatasubramanian V. Automatic generation of qualitative descriptions of process trends for fault detection and diagnosis. Engineering Applications of Artificial Intelligence. 1991;4(5):329–339.

[13] Charbonnier S, Portet F. A self-tuning adaptive trend extraction method for process monitoring and diagnosis. Journal of Process Control. 2012;22(6):1127–1138.

[14] Zhang L, Cai S, Hu J. An adaptive pre-warning method based on trend monitoring: Application to an oil refining process. Measurement. 2019; 139:163–176.

[15] Mikles J, Fikar M. Process modelling, identification, and control. Springer Berlin Heidelberg; 2007.

[16] Ghasem N. Modeling and simulation of chemical process systems. Boca Raton, FL; 2019.

[17] Frisk E. Simulation of differential-algebraic equations [Course material]. Linkoping, Sweden: Department of Electrical Engineering, Linkoping University; 2020. Available from: https://www.fs.isy.liu.se/Edu/ Courses/Simulation/OH/dae3.pdf.

[18] Martí L, Sanchez-Pi N, Molina J, Garcia A. Anomaly Detection Based on Sensor Data in Petroleum Industry Applications. Sensors. 2015;15(2): 2774–2797.

[19] Jaw LC. Recent advancements in aircraft engine health management (EHM) technologies and recommendations for the next step. In: Turbo Expo: Power for Land, Sea, and Air. vol. 46997; 2005. p. 683–695.

[20] Simon DL, Bird J, Davison C, Volponi A, Iverson RE. Benchmarking Gas Path Diagnostic Methods: A Public Approach; 2008.

[21] Sebestyen G, Hangan A, Czako Z, Kovacs G. A Taxonomy and Platform for Anomaly Detection. In: 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR). New York, USA: IEEE; 2018.

[22] Agrawal S, Agrawal J. Survey on anomaly detection using data mining techniques. In: 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems. vol. 60. Amsterdam, Netherlands: Elsevier B.V.; 2015. p. 708–713.

[23] Sadough Vanini ZN, Meskin N, Khorasani K. Multiple-Model Sensor and Components Fault Diagnosis in Gas Turbine Engines Using Autoassociative Neural Networks. Journal of Engineering for Gas Turbines and Power. 2014 5;136(9).

[24] Andersson N, Wilson DI, Germgård U, Germgard U. Validating continuous kraft digester kinetic models with online NIR measurements. Proceedings of the American Control Conference. 2002;5:3783–3787.

[25] Santos VO, Oliveira FCC, Lima DG, Petry AC, Garcia E, Suarez PAZ, et al. A comparative study of diesel analysis by FTIR, FTNIR and FT-Raman spectroscopy using PLS and artificial neural network analysis. Analytica Chimica Acta. 2005;547(2): 188–196.

[26] Skvaril J, Kyprianidis KG, Dahlquist E. Applications of near-infrared spectroscopy (NIRS) in biomass energy conversion processes: A review. Applied Spectroscopy Reviews. 2017 9;52(8):675–728.

[27] Kallestrup KB, Lynge LH, Akkerman R, Oddsdottir TA. Decision support in hierarchical planning systems: The case of procurement planning in oil refining industries. Decision Support Systems. 2014;68: 49–63.

[28] Felsberger A, Oberegger B, Reiner G. A Review of Decision Support Systems for Manufacturing Systems. In: SamI40 workshop at i-KNOW '16 October 18–19, 2016, Graz, Austria; 2016.

[29] Prasad D, Ratna S. Decision support systems in the metal casting industry: An academic review of research articles. Materials Today: Proceedings. 2018;5(1, Part 1):1298–1312.