EMERGENCE AND COMPLEX SYSTEMS
–
TOWARDS A NEW SCIENCE OF INDUSTRIAL AUTOMATION?

Heikki Hyötyniemi

Helsinki University of Technology
Control Engineering Laboratory
P.O. Box 5400, FIN-02015 HUT, Finland

ABSTRACT

Complex systems theory promises to solve all problems, big and small – but, concretely, what would such a solution mean in the first place? Specially, what would the automation environment of the future look like? The following presentation introduces these issues, trying to reach a "systemic view" of complexity.

INTRODUCTION

When looking at the field of automation and systems engineering today, it is clear that there are plenty of challenges facing us. Because of the new devices and technologies, there is an "information explosion" what comes to available process data. The process models are also becoming more and more sophisticated. How to cope with the wealth of data, and how to cope with the wealth of models?

New approaches are needed. One of the new approaches promising wonderful solutions to this kind of problems is the theory of *complex systems*. The idea of complex systems theory is intuitively appealing. We see various examples of complex systems around us, and we understand that there is something fundamentally similar beneath the surface. However, the systems we see are so multifaceted that it is difficult to see what this similarity is. In complex systems theory it is assumed that there exist some underlying very simple processes, so that when iterated massively, something qualitatively different comes out. The observed complexity is just an *emergent phenomenon*, and it is enough just to reveal the underlying simple single function to completely understand the fundamentals concerning the system and its behaviour. This is what the advocates of the New Science tell us.

This presentation will explain what the highly abstract ideas of emergence, etc., really have to do with practical automation systems – or what they might have. In addition to offering new insight into the architectures of future design/configuration/simulation environments, it turns out that there may be new "complexity-inspired" tools for implementing new schemes for parameter optimisation, adaptive control, and self-organisation of model structures. This presentation is divided in three parts:

- In the first section, an overview of the field of complex systems is presented, and the promises are reviewed; one prototypical approach to applying the complexity ideas to practical modelling problems is presented, and it is recognized that the theory is far from mature.
- In the second section, the special case of complex automation systems is concentrated on, starting from the bottom and proceeding towards increasing generality. Typically in complex systems research, the hypotheses are left on a very abstract level, with no connection to real life; now, on the other hand, not only abstract principles are presented, but a glimpse of the concrete data processing mechanisms is shown. This discussion gets more and more detailed as it continues, explaining how the proposed data-based and feature-oriented approach might offer a homogeneous view over the heterogeneity of systems.
- In the final section, the experiences are summarised. It turns out that, when struggling towards the "New Science", a plenty of deep questions emerge – questions that are by no means new. Only the answers may have changed.

ON COMPLEX SYSTEMS

It has been demonstrated by chaos theorists that mystical worlds can be constructed when strange "laws of nature" are being superimposed. In this sense, complex systems research is more like art, not being limited to any real-life constraints. However, as engineers, we are more interested in the mysteries of the real Nature around us. The key problem is to harness the emergent processes so that they would reveal something essential about the real complex systems around us.

The "New Science"

During the last year the ever so turbulent arena of complex systems has been in a total turmoil. Stephen Wolfram, a rascal in the academic circles, published his long-awaited book [21], claiming that *now it is there* – the complete theory of everything, the New Science, explaining all systems in terms of *cellular automata*.

However, the research of complex systems has a longer history, and there have been researchers exploiting this area for decades. Concepts like *Poincaré orbits*, *Lorentz attractors*, *Mandelbrot sets*, *Feigenbaumian universality*, and *Conway's Game of Life*, are familiar buzzwords from the past. Indeed, when trying to understand complex systems research, history cannot be forgotten. Typically, in sciences, it is the results that are of importance, and one can forget about the background with its detours and dead ends. Now, on the other hand, one is facing a science during its making, and everything is in turmoil. It is the complex system of researchers that can also be seen as a complex system: Every now and then something interesting emerges from the chaos.

The field of complex systems research, and approaches to attacking the challenges, are just as diverse as the complex systems themselves are. In this context, it is assumed that the kernel of complexity can be expressed as

$$x(k+1) = g(x(k)) \tag{1}$$

for some nonlinear function g. The iteration implements some kind of *positive feedback* in the system, so that the *state* x is circulated in the function. Often, if the function g is selected in a clever way, something may *emerge* from the iteration. What this emergent pattern is – by definition, these patterns cannot be predicted beforehand if only the function form g is known. In this sense, the patterns are *holistic* rather than reductionistic. Selecting the nonlinearity appropriately, the celebrated concepts like *deterministic chaos*, *bifurcations*, *strange attractors*, *fractality*, and *self-similarity* can be found within this framework.

Generally, when iterations cumulate in (1), the results become intractable if using traditional mathematics. The only way to see the patterns is to explicitly carry out the iterations. The research on complex systems is based on exploratory experimenting, and there are more simulations than analytic proofs. The researchers are trying to capture the essence of complex systems by constructing different kinds of environments where something interesting is found. Complex systems research is synthetic rather than analytic – for example, in artificial intelligence (AI) research one is going from "intelligence as we know it" to "intelligence as it could be". If enough material is gathered, perhaps the principles of *universal intelligence* can be seen.

It has been claimed – not only by Wolfram – that standard mathematics cannot capture the essence of emergent phenomena. But mathematics is just another word for logical thinking – the risk is that without consistent conceptual tools everything is just handwaving. Is the New Science just a form of "ironic science", as John Horgan has put it [6]: The field of chaoplexity is full of promises, with much less concrete results. Scientific salesmen have been selling miracle cure for all problems. All possible breakthroughs have already been distrained beforehand.

Wolfram says that old mathematics has to be forgotten altogether. True, the cellular automaton model is mathematically unpenetrable. But it may be that the unanalyzability is the property of the modelling method rather than the general property of complex systems themselves. There always exist various approaches to modelling of systems – the cellular automaton model is just one of the alternatives. Most probably complex systems do not become simple by applying any simple-minded magic trick, but what one can perhaps achieve, is new kinds of ways of seeing the structures of systems, so that the complexity could be divided into better manageable parts in fresh ways.

It is easy to make unsubstantiated hypotheses, whereas the main challenge is to combine the new ideas with the prevailing conceptual and methodological frameworks. Not all magnificent old theories can be abandoned – the test for the new ideas is how they can be integrated with the existing body of understanding. How to find methods for mastering the complex processes, and tools for controlling the emergent patterns? How to reach some kind of "holistic mathematics" for capturing the essence of complex systems?

Modelling Tools

Even if Stephen Wolfram claims that all complex systems *are* cellular automata, one should admit that one only tries to construct *models* for such systems. This means that one can avoid metaphysical questions – what the reality really is – and concentrate on more fruitful issues: For a model, it is allowed that some details are abstracted away, and only some relevant aspects are concentrated on. One of the roles of a model is to make the complex system better comprehensible. In this respect, the Wolfram's approach (reducing everything back to elementary units or "cells", explaining phenomena using very low-level) may be a good theory, but it is bad modelling.

To reach understandable models, special tools for that purpose are needed to reach higher-level view of a system. New ways of thinking necessitate new concepts and new structures for connecting these concepts. In the computer-age, the new concepts are employed by specialised *formal languages*. When they are implemented, the resulting programming languages reflect the ways how the world is being conceptually put together. To see the continuum from today's tools to the future, it is instructive to briefly look at the old languages for representing model prototypes.

The description languages have always been designed for making it easy to express things in the assumed structural framework. Roughly speaking, the history of programming languages started with a *centralised view* often resulting in "spaghetti code" (Basic, Fortran). After that, procedural languages were introduced, where blocks of code were clearly distinguished (Algol, Pascal). When the systems being modelled became larger, the decentralisation of control within the formalisms continued, and different kinds of object formalisms were developed (C++, Java). Today, it seems that in the *agent paradigm* [19] this decentralisation is being taken to its extreme: No centralised control any more exists. In the special-purpose formalisms, for example in simulation languages the same tendencies can be seen: First, the systems were defined by hard-wiring, whereas today all simulation/configuration environments support hierarchic, modular model construction.

There are also philosophically deeper motivations for the practical programming languages. The current object-oriented formalisms are an implementation of the Simonian *hierarchies* [20], being further based on Aristotelian taxonomies with strict organisation among the entities. The theories beyond agents and agent networks are supported, for example, by the studies by Albert-Laszlo Barabasi [1], and also by John Doyle and collaborators [5], where they show that new approaches are needed to master the robustness properties in decentralised but mutually interconnected devices. As an extreme case of decentralisation and local interactions one has the cellular automaton paradigm.

It is not reasonable to assume that the current tools would be final – but it is extremely difficult to see what kind of functionalities should be supported by the new tools: "If we knew the future, we would already be there." Some intuitions can perhaps be gained when looking at concrete applications – and, indeed, such studies will be presented later in this paper. Before that, an example of today's approaches towards applying the new concepts is presented.

Example: Application of Fractal Theory

The field of complex systems research is a rather complex system itself – but some kind of patterns have already emerged. One of the most interesting and promising ideas is related to *self-similarity* and *fractality*. First, basic ideas beyond fractal structures need to be studied (for example, see [4]).

Dimension of a geometric construction is an age-old concept; traditionally, only integer dimension values can exist. However, some objects cannot naturally be classified in such a way; there seem to exist structures that fall somewhere "between dimensions". *Fractal dimension* is an extension of this dimensionality concept towards fractional values:

$$D = \frac{\log N}{\log S} . \tag{2}$$

Here, N stands for the *number of self-similar substructures*, and S represents the *scaling factor*. That is, assuming that a structure can be reconstructed using various smaller but equally shaped substructures, and this self-similarity continues at different scales, the above definition can be applied. For example, a line segment can be reconstructed from two segments having half of the original length; this means that in this case $D = \log 2 / \log 2 = 1$ (note that the line segment is self-similar in *all scales*; dividing it to three identical parts, there would still hold $D = \log 3 / \log 3 = 1$). A square can be reconstructed from four squares having sides that are half of the original size, so that $D = \log 4 / \log 2 = \log 2^2 / \log 2 = 2$. The fractal dimension equals the traditional dimension if it is defined. On the other hand, look at the self-similar structure in Fig. 1: This "Sierpinski Triangle" can be reconstructed by combining three similar constructs having half of the side length, so that now $D = \log 3 / \log 2$, the dimension being somewhere between 1 and 2 (compare this triangle pattern to the images shown in [21], having very different origin)!

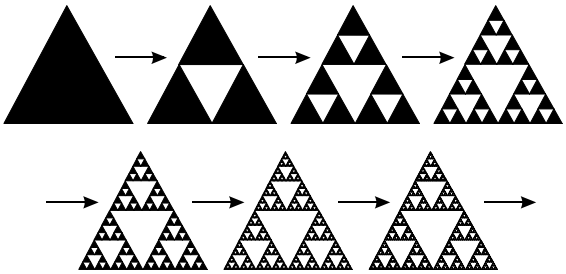

Figure 1. Construction of a self-similar pattern

This kind of self-similarity and fractality has been observed in many natural forms – indeed, it has been said that fractality is the Geometry of Nature. For example, leaves, coastlines, etc., exhibit this phenomenon: The same patterns recur in different scales. More generally, it has been claimed that in all structures that have evolved naturally, various low-level structures interacting and self-organising, this kind of self-similarity emerges.

What are the consequences of this self-similarity? When the expression (2) is written in another form

$$\log N = D \cdot \log S \ , \tag{3}$$

it turns out that for a self-similar structure, the relationship between the quantities S and N, when plotted in a log-log scale, is *linear*, the slope being determined by the fractal dimension. This *power law* pops up in seemingly very odd places. Indeed, before fractal theory, it was known as *Zipf's law*: It had been noticed that city sizes within a country, word frequencies in a language, etc., seem to follow this strange law. The power law behaviour makes it easy to estimate the fractality beyond complex systems, and it makes it possible to estimate some of their properties on a higher, more abstract level. The properties of complex systems have been studied within this framework using names like *self-organised criticality*, *phase transitions*, *edge of chaos*, and *highly optimised tolerance*.

And look at Fig. 2: Even if the schematic illustration is oversimplified, it is evident that the cascade control structures within a complex industrial plant are self-similar. It can be assumed that the number of coupled control loops still increases when, for example, the control is distributed and the agent paradigm is fully exploited, meaning that system fractality will increase. Other technical examples of fractal structures include electrical and Internet systems: Both of these environments consist of large networks having self-similar sub-networks within. Thus, power outages and data transfer frequencies and rates should follow the power law – and, indeed, this kind of behaviour has been observed also in practice.

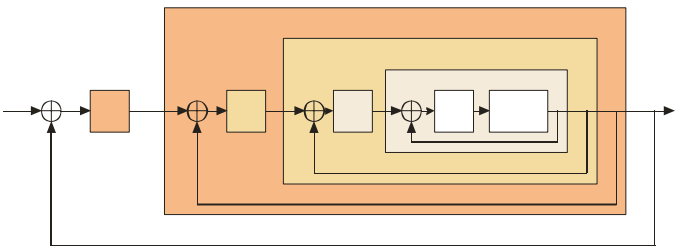

Figure 2. Cascade control structure

Because Nature seems to work in this fractal way, because such structures have survived the struggles of evolution, is it not reasonable to try and mimic the same principles also in artificial systems? And, indeed, the robustness against unmodelled phenomena seems to be enhanced when redundancy in the system is distributed so that the fractality in the system is reserved at different levels. In different environments, the "yardstick" S can have different interpretations: For example, in an automation system, the quantity that can be designed to obey the power law might be the number of control loops in different levels, or it might be the lengths of sampling intervals in the loops.

Would this framework of self-similarity be a good starting point for a general theory of large-scale automation systems? No, not really – it has to be noted that the self-similarity approach remains on a *too high* level: There are no connections to down-to earth techniques currently applied in the field (for example, take *parameter estimation*: How could the above ideas be applied there? Do they offer any fresh intuitions?). The self-similarity idea is too qualitative and descriptive as an approach to be adopted as a general framework, and more quantitative tools are needed.

TOWARDS A NEW PARADIGM

Aiming towards a totally new science of complex systems is probably too ambitious. But, as Thomas Kuhn put it, there are *paradigm shifts* within a science – things are seen from another point of view, and new conceptual tools are introduced. What could such a shift look like? In this section, a simple scheme is presented – the main objective is only to illustrate the possibilities there exist. It is demonstrated how systemic thinking may help in understanding complex systems, and, on the other hand, it is visualised what kind of new challenges the new, data-centered world is issuing to general system theory [3].

Levels of Abstraction

It seems that when there are high numbers of lower-level entities together, the appropriate way of looking at the whole system may change altogether – the relevant variables are changed, and it seems that even the basic paradigm may become obsolete. As an example, study the *modelling of gases* at different levels of abstraction:

- Elementary particles of the gas behave *stochastically* (as *quantum theory* has to be applied)
- Atoms behave *deterministically* (Newtonian ideal gas model, based on *classical mechanics*)
- Atom groups behave *stochastically* (*statistical mechanics* with velocity distributions, etc.)
- Large volumes behave *deterministically* (state being described in terms of pressures and temperatures)
- Still larger volumes behave *stochastically* (turbulence and fluctuations becoming acute)
- Perfectly stirred volumes behave *deterministically* (ideal mixer model, behaviour being governed by first-order ordinary differential equations).

This does not mean that the underlying laws would change; in principle they still apply. It is only that the lower-level tools are no more the most economical ones for discussing the relevant phenomena. It is like trying to understand a complex program when only the machine code is seen. One of the main roles of a model is to help the user understand the system being studied, and to facilitate this understanding, the modelling approach must be as transparent as possible. The total complexity of understanding a system model is some kind of a sum of the complexity of the applied methodology, and of the complexity of the system itself. When the system truly is complex, one should pursue towards as simple models as possible.

The *component level* of ideal mixers, etc., is what is now being exerted for describing complex systems. However, when looking at the *plant level*, there are dozens of mixers, and it has been observed that the deterministic, exhaustive approach, trying to exactly keep track of each subsystem, becomes very cumbersome. The number of free parameters is the sum of the parameters in the components. It seems that the complete industrial plant cannot be mastered when it is studied at the level of the individual devices. Looking at the list above – is it not a good assumption that the next, higher level of abstraction should again be *stochastic?*

World of Statistical Data

How to get rid of the details, how to reach the higher level of abstraction? The key is presented in Fig. 3: Rather than looking at individual realisations of input and output signals u and y, one concentrates on the *qualities* of the system behaviour. Depending on the *qualifiers* (parameters, set point values, etc. – any variables that may help in explaining the system operation) the observed system behaviour changes, and one can concentrate on the truly interesting system properties. Of course, being functions of arbitrary signal realisations, the results are more or less random; assuming that there really is some connection between the qualifiers θ and the qualities q, these dependencies can be seen in the statistical data. Extensive *Monte Carlo simulations* with slightly varied operating conditions should deliver the necessary information, spanning the local environment in the parameter space.

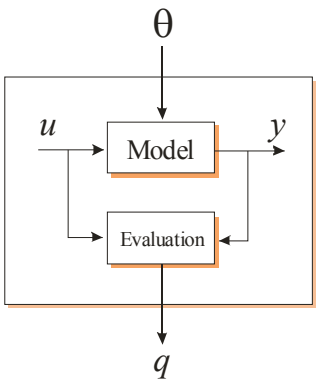

Figure 3. A more abstract level of looking at a dynamic system

The presented scheme is flexible: It does not matter whether the underlying object producing the signals is an actual real system or some kind of computational model. It is only assumed that there is some means of modifying the system behaviour, so that there is a set of adjustable qualifiers; it is also assumed that the input-output data can be measured, and the qualities can be evaluated. It is up to the application developer to select the qualities to be studied, and formulate expressions for assessing the quality of the observed behaviour appropriately.

When studying the new way of looking at systems from a distance, there are various items that need to be recognised:

- Homogeneity of representations is reached, no matter what the underlying devices or models are like.
- Simplification is reached: Dynamic models become static, individual signals can be abstracted away.
- However, new problems also emerge: Because of the high dimensionality of the data, new kinds of tools for *multivariate statistics* are needed (for example, see [11]).

In [12], it is illustrated how the model or controller parameters can be optimised applying multivariate regression techniques in the above framework. In what follows, however, the homogeneity of the data will be assumed and utilised; rather than having separate vectors for qualifiers and quantities, there is just one data vector for each observation sample:

$$\chi = \left(\frac{q}{\theta} \right). \tag{4}$$

That is, all relevant data is collected in the same vector structure. After appropriate scaling operations, etc., it seems that the multi-faceted structures of the heterogeneous device-level world have been transformed into homogeneity. But this new world of data is "without form, and void" – what kind of new, higher-level structures to introduce to master this wealth of data?

Introducing Structure

To continue, assume that among the data in the vector χ all information that may be relevant at that level of modelling is included: Actual process parameters, set point values of signals, or even time-series sequences of variables. Further, assume that there exists a large amount of vectors χ, so that something about the statistical distribution can be said.

First, let us concentrate on a simple case – assume that the data is *unimodal*. Assuming that the variations in data are normally distributed – as the *central limit theorem* motivates – the data can be expressed by a *multivariate Gaussian (normal) distribution*. The Gaussian distribution offers us a nice mathematical framework: First, it turns out that the distribution is uniquely determined by its center and its *covariance matrix*. The covariance matrix reveals how much the distribution extends in different directions; the eigenvectors of this covariance matrix are called *principal components*. Most of the information (or, at least, most of the variation) in the data is captured in the subspace that is determined by the most significant principal components.

Another crucial observation is that, as it turns out, the optimal models (in the maximum likelihood sense) involving normally distributed variables are *linear*. This linearity is the starting point we are adopting here.

The model for data is based on *linear features*. This means that the observed pattern χ can (approximately) be represented as a linear combination of feature vectors C_i, where $1 \le i \le N$, the corresponding weighting factors being x_i. The number of feature vectors, N, is typically lower than n, the dimension of χ:

$$\chi = x_1 \cdot C_1 + \cdots x_N \cdot C_N . \tag{5}$$

This can also be written as

$$\chi = C \cdot x , \tag{6}$$

where C contains feature vectors as columns, and x contains their weights in vector form. To exactly reconstruct the data vector, the rank of C should equal the dimension of the data vector x. However, it often turns out that the degrees of freedom in the data are (essentially) below this upper bound. Applying some clever feature extraction schemes, the columns in C can be determined so that they span the data space so that the vector χ can be represented by the lower-dimensional vector x. For example, in the case of unimodal data, the features can represent the directions of the most significant principal components. Some practical examples of "featuring a system" are presented in [18].

In mathematical terms, the columns of C determine the (not necessarily orthonormal) *basis vectors*, or coordinate axes in the lower-dimensional subspace, whereas the elements in x determine the corresponding

coordinate values. In intuitive terms, on the other hand, χ is a vector containing the observed *external pattern* of variables, whereas x in an *internal representation* of it. Indeed, x can be regarded as a (generalized) system *state*. When the dimension of x is lower than that of χ, it can only represent a subset of all possible variable combinations in χ. Assuming that C has been appropriately constructed, so that it contains some kind of correlation structures observed in χ, this dimension reduction can be carried out so that only *relevant* information is retained: The data is often corrupted by noise, and ignoring that disinformation is a clever strategy.

When implementing a model for Gaussian distribution, columns in C can be used to store the vector that represents the center point of the distribution; similarly, columns in C can represent the linear features spanning the degrees of freedom within the distribution. Assuming that there are N degrees of freedom in the data, one needs N+1 variables to span this space. One of the variables in x is always 1, standing for the center point, whereas the rest N variables can have arbitrary values. This means that the feature representation is uniform for all data constructs that are needed to represent Gaussian clusters.

To be truly potential, the feature-based approach should make it possible to implement also the traditional ways of modelling of "simple" systems in the new framework. Indeed, for example, to see how the problem of representing linear d'th order discrete-time dynamics of the form

$$y(k) = a_1 y(k-1) + \cdots a_d y(k-d) + b_1 u(k-1) + \cdots b_d u(k-d) \tag{7}$$

can be accomplished in the new framework, one has to define the $2d$+1 -dimensional data vector as

$$\chi(k) = \left(y(k) \mid y(k-1) \quad \cdots \quad y(k-d) \mid u(k-1) \quad \cdots \quad u(k-d) \right)^T . \tag{8}$$

To capture the linear dynamics, the matrix C has to contain only $2d$ feature vectors. That is, the internal representation of the dynamics, or x, has dimension one less than the data vector has; this is due to the fact that the equation (7) reduces the degrees of freedom by one. After the columns of the matrix C have been appropriately determined, all combinations of variables – and only those – that fulfill the constraint (7) can exactly be presented by the matrix-form model (6). The "parameters" of this model are typically not directly connected to individual time series variables, but their effects (if interpreted in the time domain) are distributed in a very different way, reflecting the underlying dependencies between the signal samples. From the identification point of view, there are nice benefits available here: The parameter estimation task is now implemented in the multivariate statistic setting. If the structural identifiability properties of (7) are poor, meaning that one cannot detect the effect of individual parameters in the data, the least relevant "parameters" can be dropped when constructing C (when applying, for example, principal component analysis PCA).

As compared to the original input-output formulation of the dynamics (7), the new model with no explicit input or output variables is seemingly less oriented towards systems engineering applications – however, the homogeneous data representations can be restructured in the causal form by applying the idea of *associative regression*: One matches the known variables (inputs) against the model, determining the internal representation x that best matches the partial data, and reconstructs the unknown variables (outputs) by mapping this internal representation back to external representation χ, the unknown entries being automatically filled in (see [14]).

Because parameter estimation is now based on PCA type subspace-oriented schemes, one does not have to be so careful about explicitly minimising the data dimension. For example, if there is some additional information available, it may be motivated to introduce augmented data vectors of the form

$$\chi(k) = \left(y(k) \mid y(k-1) \quad \cdots \quad y(k-d) \mid u(k-1) \quad \cdots \quad u(k-d) \mid z_1(k) \quad \cdots \quad z_m(k) \right)^T . \tag{9}$$

If it turns out that the external variables z_i are correlated with the variable y, these dependencies are included in the data structures, thus extending the range of the model: For example, the external variables can tell something about the load and its variations; when this information is included in the model, fast adaptation to load disturbances can be implemented as a kind of gain scheduling scheme, applying feedforward disturbance compensation. Such issues are further elaborated on in [14], where "life-like" control is implemented in the presented framework.

Emergent Patterns

Unfortunately, when studying a complex system, the above unimodality assumption cannot be made. In general, when studying the properties of a distribution, one is facing a *data mining* problem. In our case, however, it turns out that the problem is not so bad: One is not totally lost in the jungle of structureless data. There exist some structural properties that apply, and special assumptions about the distribution properties can be made [9].

Feature extraction and *pattern recognition* have been extensively studied in the field of computational intelligence. What makes the current task somewhat different, is that the assumed underlying model structures at the lowest level are unimodal and linear, and efficient statistical tools can be applied. There exist various ways of determining the linear feature vectors so that the correlation structures in the data are captured. The basic approach is *principal component analysis* (PCA): The features are the most significant eigenvectors (as measured by the corresponding eigenvalue) of the data covariance matrix (for example, see [2]). This method is mathematically efficient and there are various technical benefits – for example, the feature basis determined by the principal components is automatically orthonormal, making it easy to match the data against the feature model. However, mathematical efficiency does not mean *physical* relevance. Whereas PCA is good approach for unimodal data, for *multimodal* data it creates some kind of average model that is really good in none of the modes. As motivated by the *independent component analysis* (ICA) research, the physically meaningful features typically are not orthogonal [7]. It can be claimed that *sparse components* are best suited for modelling of natural, multimodal data representing a *Gaussian mixture model*.

To conclude: A natural data distribution is multimodal, consisting of a mixture of various mutually exclusive Gaussians. This means that the appropriate model is *piecewise linear*, consisting of linear substructures that are spanned by the individual data clusters each having separate normal distribution.

It can be claimed that *sparse coding* is a good approach to capturing mixture models. In sparse coding one does not try to minimise only the matching error, but also the number of *simultaneously active* components (features). That is, only a subset of elements in x have non-zero values. This assumption is motivated in Fig. 4: Assume that there are various mutually exclusive Gaussian clusters present in the data, meaning that a data vector can be produced by just one of the subdistributions alone at any given time. The C matrix may contain more columns than what is the dimension of data, because the features for each of the subdistributions need to be listed there; the high number of columns in C is compensated by the fact that only a subset of those vectors are simultaneously utilised, meaning that uniqueness of x can be reached.

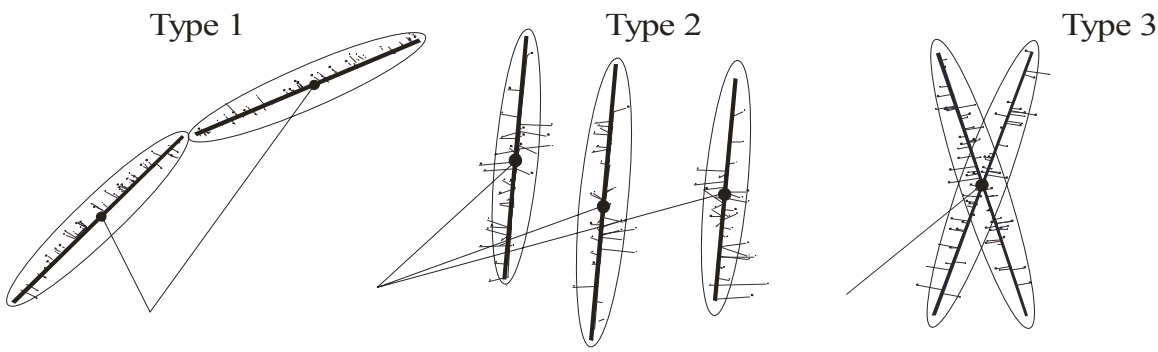

Figure 4. Classes of "natural" patterns typically found in data

The different basic types of Gaussian mixtures, as shown in Fig. 4, can be motivated in a rather natural ways (as presented in [9]):

1. *Continuous nonlinearity.* Smooth nonlinear functions can be approximated by locally linearising the function and using the "nearest" submodel to represent each sample. When nonlinear dynamic systems are being studied, this kind of linearisation around the operating points is a standard technique.
2. *Clustered data.* Assuming that there exist, for example, different operating modes or fault conditions, or assuming that the process operators have different ways of controlling the process, there typically exist distinct pile-ups of data, each of them being (more or less) Gaussian.
3. *Independent components.* The new trend in statistical analysis is that of concentrating not only on uncorrelatedness but on independence [7]. Without going into details, it seems that independent components are capable of nicely capturing physically relevant phenomena.

All of the above structural phenomena are rather natural and easily motivated; no conceptually very new ideas need to be introduced. This kind of fit between old and new approaches is, of course, a nice coincidence, making it easier to (perhaps) adopt the new ideas. For a concrete example, let us briefly study an example from mineral processing industry, *calibration of an X-ray fluorescence analyser*. It turns out that all of the above structural phenomena can be detected in such a system; it also seems that the proposed approaches may immediately suggest some new innovations for enhancing the device functioning.

The X-ray analyser is a device that uses X-rays to determine the contents of a sample by measuring and analysing the reflected intensity spectra. Each atom has typical ways of emitting radiation; inspecting the spectrum peaks gives evidence of the mineral contents in the sample. The spectrum intensity readings have to be calibrated, matched against the real sample contents as determined in a laboratory. The correspondence between the intensity channel measurements and the atomic contents is *basically linear* – however, there exist also *continuous nonlinearities*, mainly caused by the density changes in the slurry. Typically, calibration is carried out by constructing a linear regression model; because of the theoretical limitations of linear regression, the robustness of the model is not very good; not all available data can be included in the model, even if it might be potentially informative. As shown in [10], multivariate methods can offer considerable benefits in calibration. What is nice, the latent variable approach of the more sophisticated multivariate methods is completely compatible with the proposed feature-based models: The latent subspace axes correspond to the features. Looking closer at the properties of the analyser data, one can notice that the atom counts cannot vary independently; it is the chemical compounds that determine the *independent components* within the data. Modelling the observations in terms of the real underlying minerals can make the data matching process more robust. Note that such independent components really are non-orthogonal: Various minerals can contain the same elements (unfortunately, the lighter elements cannot usually be detected in the intensity region that is being monitored; this means that the minerals cannot necessarily be distinguished based on the analyser data alone). Different ore types that are coming from different parts of the mine, for example, typically have different properties, giving raise to *clusters* in the data. When the ore properties change, new calibration campaigns need to be carried out; however, typically the same ore types repeat, and it is not clever to forget about old calibration models altogether. It turns out that the mixture model offers a nice framework for storing past information for future reference, waiting there for the similar operating conditions to recur. This all means that the presented feature model framework should be well suited for modelling the analysis process, that is, for calibrating the device.

Today, there are efforts towards making the X-ray analyses more accurate by exploiting the *mass balance* relationships; that is, *data reconciliation* is possible if one can measure all flows and concentrations in and out from a (sub)system. However, such redundant measurements are not typically available, and different kinds of simplifications are necessary: For example, one typically only employs static balance models. Often, the same analyser is used to measure various slurry flows simultaneously, shuffling between the flows. It turns out that when looking at the overall analyser and all data therein as a single system, new interesting approaches are possible. The flows come from different parts of the process, meaning that the properties of the flows are related to each other, reflecting the outside process structure and its dynamics. Even though the information is only partial and there are delays, the relationships can be captured if the statistical modelling tools are efficient enough. The analyser can perhaps carry out intelligent sensor fusion (semi)automatically, constructing the dependency models between the flow variables according to the observations. Automatic adaptation in an unstructured environment is possible – the X-ray analyser can become a truly *smart* device, acting intelligently in its narrow world determined by its sensors.

Expressive Power

Looking at the proposed feature model, it is clear that merely linear representations are too weak to capture the real-life complexity – when linear structures are combined, the overall structure is still linear. Specifically, the linear representations are naturally *dense*, so that the above assumed sparsity, even though it is possible, it is not explicitly supported. What kind of additional nonlinearity to introduce to make more complex representations possible, and to facilitate emergence of sparsity? Again, it is assumed that the basic underlying structure is linear (affine), boosted with nonlinearity, so that

$$\chi = f(Cx + b)\,. \tag{10}$$

This means that one first applies the linear (affine) mapping, as before, and only after that the nonlinearity is applied. When selecting the nonlinearity, a Pandora's box is opened; it needs to be recognized that now this discussion is becoming (even more) speculative. Anyway, getting forward and seeing (one view) of the big picture necessitates some radical assumptions. Let us study one example of all available nonlinearities – it seems that a nice compromise between simplicity and expressional power is the *cut function*:

$$f_i(\xi) = \begin{cases} \xi_i, & \text{if } \xi_i > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

This function is calculated elementwise: If some entry in vector ξ is negative, it is simply zeroed (see Fig. 5). Because of this, all variables are non-negative – meaning that there is a close connection to the theory of *positive systems*. The positivity assumption is well motivated in many physical environments; a pragmatic way to motivate the above selection (11) is that it can be claimed to be (in the sense of *Occam's razor*) the *simplest possible* nonlinearity (even though this apparent simplicity is misleading – see later).

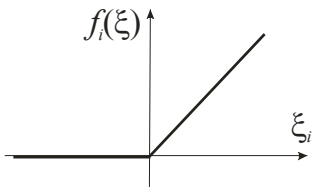

Figure 5. Nonlinearity that makes the variables strictly non-negative

Note that no functionality is lost because of the assumed nonlinearity: Strictly linear mappings can be implemented in this framework when one allocates *two* positive variables for each degree of freedom, one for positive and the other for negative direction. With the cost of higher dimensionality, linear (affine) mappings can be implemented (in the first quadrant):

$$\chi = f\left(\left(C \;\vdots\; -C \right) \begin{pmatrix} x^+ \\ \hdashline x^- \end{pmatrix} + b \right). \tag{12}$$

The cut function promotes structure and emerging sparsity, because any variable can be explicitly zeroed by adding a large negative value to it. This means that any substructure can be deactivated by using some of the x_i's as control variables – this control mechanism is the key to managing larger models and flexible hierarchies, consisting of various mutually exclusive substructures (see [15]). It is evident that efficient conceptual tools are needed to master the complicated control structures in such a simplistic framework – this question is elaborated on in next section.

Increasing Functionality

As shown above, reconstruction of the external pattern χ is straightforward when the internal image (or system state) x is given; the difficult task is to determine x when only χ is known. And how to determine C and b? Typically, x is much lower-dimensional than χ, and some kind of clever data compression needs to take place in this process: How to make sure that it is the relevant information that only remains?

More functionality is clearly needed – the formula (10) alone is not flexible enough for us to reach something truly useful. How to implement not only simple mappings but also more complex tasks, still assuming that $f(Cx+b)$ is the *only* functionality there is available? In the spirit of complex systems, using the available functionality one can define a recursion of the form (1):

$$x(\kappa+1) = f\left(Ax(\kappa) + b \right), \tag{13}$$

or, more specifically,

$$x(\kappa+1) = f\left(Ax(\kappa) + B\chi + c \right). \tag{14}$$

Given the data χ, it is now assumed that the corresponding analysis result, or the internal state x emerges when the iteration is carried out long enough, starting from some initial state $x(0)$:

$$x = \lim_{\kappa\to\infty} x(\kappa). \tag{15}$$

The key question now is: How can the high-dimensional observation data be matched against the feature-form system model so that the internal state could be found, where only the relevant phenomena are represented? To determine A, B, and c in (14) so that the emergent pattern would be some relevant internal state of the system, it is *mathematics* that is needed. Indeed, it turns out that mathematics is not only the way to abstractions, but it is also the way to reach something concrete. *Language of holism is mathematics* – to see this, first study a general, quadratic cost criterion:

$$J(x,\chi) = \frac{1}{2}\cdot x^T H_1 x + \frac{1}{2}\cdot\left(\chi - Cx\right)^T H_2 \left(\chi - Cx\right) + h_3^T x. \tag{16}$$

It is assumed that H_1 and H_2 are symmetric weighting matrices. Minimisation of this expression with respect to x can be carried out, for example, by applying the steepest descent algorithm, taking successive steps of length μ along the negative gradient direction:

$$x(\kappa+1) = \left(I - \mu H_1 - \mu C^T H_2 C \right)\cdot x(\kappa) + \mu C^T H_2 \cdot \chi - \mu h_3. \tag{17}$$

As illustrated in [13], it may be motivated to introduce *non-positive definite* weighting matrices. This means that there does not exist a finite solution for the optimisation problem. This problem can be avoided,

if the growth of the variables is appropriately limited. If the variables are restricted to the positive domain, one can write the optimisation algorithm as

$$x(\kappa+1) = f\left(\left(I - \mu H_1 - \mu C^T H_2 C\right)\cdot x(\kappa) + \mu C^T H_2 \cdot \chi - \mu h_3\right), \tag{18}$$

where f again is the cut function. If one defines new data structures as follows

$$\begin{cases} A = I - \mu H_1 - \mu C^T H_2 C \\ B = \mu C^T H_2 \\ c = -\mu h_3, \end{cases} \tag{19}$$

one can see that the iteration is in the form (14). In this way, one just has to express the model matching problem as a quadratic optimisation problem, and one can readily determine a process from which the solution emerges. This means that *one can see the emergent patterns before they emerge from iterations*. Rather than having to study the complicated nonlinear dynamic processes, one can directly concentrate on the emergent static pattern, the optimality criterion.

There now exists an efficient machinery for making interesting patterns emerge in the iteration process. As an example, study the determination of the "coordinates" x when the data vector χ and the feature matrix C (not necessarily orthogonal) are given. It needs to be noted that the matching problem (or actually matrix inversion) is represented by the following cost criterion

$$J(x,\chi) = (\chi - Cx)^T(\chi - Cx)\,. \tag{20}$$

This means that one simply selects $H_1 = \mathbf{0}$, $H_2 = 1/2\, I$, and $h_3 = \mathbf{0}$ in (16) and (19). This criterion does not at all emphasise the variable sizes; to exactly match the data with the Gaussian distribution, applying the a posteriori log-likelihood criterion one can write a more sophisticated criterion that is again of the assumed form (16), if one neglects the a priori cluster probability p that remains constant within one submodel:

$$J'(x,\chi) = \frac{1}{2}\cdot x^T \Sigma_x^{-1} x + \frac{1}{2}\cdot(\chi - Cx)^T \Sigma_e^{-1}(\chi - Cx) - \ln p\,. \tag{21}$$

Here the Σ's are the covariance matrices of the latent variables and the errors, respectively, measuring the data match as seen from *inside* the Gaussian model (Hotelling's T^2 statistic) and from *outside* the model (Q statistic). This basic solution can also be easily extended: For example, if one of the feature vectors, say, C_1, has a special role as being the (non-scalable) cluster center vector, there must all the time hold

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}\cdot x = 1, \tag{22}$$

meaning that $x_1 = 1$. This kind of constrained optimisation problems (quadratic criterion with a set of linear constraints) can generally be expressed as

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2}\cdot x^T H_1 x + \frac{1}{2}\cdot\left(\chi - Cx\right)^T H_2\left(\chi - Cx\right) + \eta^T x \\ &\text{when} \qquad \Gamma\cdot x = \gamma. \end{aligned} \tag{23}$$

Using the Lagrangian formulation, one can write the Hamiltonian

$$\bar{J}(x,\chi) = \frac{1}{2}\cdot x^T H_1 x + \frac{1}{2}\cdot\left(\chi - Cx\right)^T H_2\left(\chi - Cx\right) + \eta^T x + \lambda^T\cdot\left(\Gamma x - \gamma\right). \tag{24}$$

Differentiating, setting the derivative to zero, and using the expression $\Gamma x = \gamma$, one can solve for the vector of Lagrangian multipliers:

$$\lambda = \left(\Gamma\left(H_1 + C^T H_2 C\right)^{-1}\Gamma^T\right)^{-1}\cdot\left(\Gamma\left(H_1 + C^T H_2 C\right)^{-1}\left(C^T H_2 \chi - \eta\right) - \gamma\right), \tag{25}$$

and, further, one can notice that the constrained optimisation problem (23) can be formulated without constraints in the form (16) when one selects

$$h_3 = \eta + \Gamma^T\left(\Gamma\left(H_1 + C^T H_2 C\right)^{-1}\Gamma^T\right)^{-1}\cdot\left(\Gamma\left(H_1 + C^T H_2 C\right)^{-1}\left(C^T H_2 \chi - \eta\right) - \gamma\right). \tag{26}$$

This can be optimised using (14) when

$$
\begin{cases}
A = I - \mu H_1 - \mu C^T H_2 C \\
B = \mu C^T H_2 - \mu \Gamma^T \left(\Gamma \left(H_1 + C^T H_2 C \right)^{-1} \Gamma^T \right)^{-1} \cdot \Gamma \left(H_1 + C^T H_2 C \right)^{-1} C^T H_2 \\
c = \mu \Gamma^T \left(\Gamma \left(H_1 + C^T H_2 C \right)^{-1} \Gamma^T \right)^{-1} \cdot \left(\Gamma \left(H_1 + C^T H_2 C \right)^{-1} \eta + \gamma \right) - \mu \eta.
\end{cases}
\tag{27}
$$

This all means that one needs to express one's modelling case as a (constrained) quadratic optimisation problem, and an emergent process can readily be written with the longed-for solution as its emergent pattern. It turns out that the presented iteration (14) is very versatile. As shown in [13], *forward* and *backward inference*, and *learning by examples* can be implemented; the highest valued element of a vector can also be determined, making it possible to distinguish between mutually exclusive clusters. Indeed, as shown in [16], *all algorithms* can also be implemented in the proposed framework; using appropriate tools, the matrix A can be written so that truly anything can be realised in this extremely simple framework (see Fig. 6)!

All of the above functionalities can be implemented in the same matrix formalism – and various independent processes can be implemented using a single expression of the form (14) if the dimension of x is high enough. This approach facilitates not only horizontal but also *vertical* homogeneity of system models: In the higher-level models, the same piecewise linearity assumptions still apply. As shown above, an individual device can be modeled; properties of subprocesses and the whole process can be expressed in the same framework; and on the highest level of production management, quadratic or linear criteria are optimised, subject to linear constraints on lower-level variables.

However, it is not necessarily clever to implement all substructures in the same, clumsy, centralised structure; it is better to model independent entities separately. One can explicitly impose hierarchy among the matching subprocesses, using the previous level x as a part of the next level χ. To make this feasible, higher-level control of elementary pattern matching processes is needed to implement models for more extensive systems. Indeed, a simulation environment in Matlab Simulink is being constructed, and a representation formalism is being developed for conceptually capturing the phenomena observed in large-scale systems (see next sections).

$$
A = \begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&-1&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&-1&0&1&0&0&0&-1&0&0&0&0&1&0&0&0&0\\
0&1&0&1&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&-1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&-1&0&0&0&0&-1&1&1&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0\\
1&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&-1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&-1&0&0&0&0&0&0&1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&1&1&0&1&0\\
0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&-1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&-1&0&0&0&0\\
0&0&-1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}
\quad \text{and} \quad
x(0) = \begin{pmatrix}
0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\2\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0
\end{pmatrix}
$$

*Figure 6. Assume that in (13) the data structures are selected so that b = **0** and A is defined as shown above, and the iteration starts from x(0), as also shown above. It turns out that always when there holds $x_{15} = 1$, there is a prime in x_{20}; when the process continues, arbitrarily large primes are produced. How is this possible? In principle, there is nothing more astonishing here than in fractals, where, too, infinitely complicated patterns are produced by finite-sized functions – now the emergence process just has been "cultivated". This is possible because the above iteration exactly corresponds to an algorithm that has been programmed to produce the sequence of primes (see [16])*

Alternative Approach

Whereas the basic approaches are seemingly sound – representation of the system information in the "homogenised" statistical framework, and structuring the data in terms of features – the above discussion may go too far in a somewhat strange direction. Specifically, the assumed nonlinearity is highly controversial; let us inspect another approach to integrating nonlinearity on top of the linear structures.

Above, the main motivation beyond the assumed nonlinearity was *simplicity*, in the spirit of complex systems research: All levels in a system model were implemented as patterns emerging from iteration of simple functions, and the whole system model could be implemented as a single huge homogeneous matrix iteration. The structure contained minimum amount of external control and a priori structure; it was assumed that somehow the structure would automatically emerge. However, it may be reasonable not to be too ambitious. The modeling tool should not be an end in itself; it is the "customer that is always right", and one should be prepared to do compromises. Compromising with generality in the above data-oriented modeling view, or explicitly introducing some internal structure with "locally centralised" control, it seems that more comprehensible models that are better compatible with traditional modelling methodologies are perhaps available.

When modelling real systems, it turns out that it is not only *one* mixture model that suffices; in practice, it seems that each device is best described by a mixture model of its own. In a large-scale system, there are mixture models on different levels, the lower-level ones supplying the higher-level ones with appropriate data. Whereas there are close interactions between variables within a single mixture model, the connections between mixture models are rather loose. It is between the mixture models where the Simonian "Almost Decomposability" and "Empty World" hypotheses [20] apply – however, the Simon's original ideas can be boosted with the modern understanding of fractality. Whereas the mixture models seem to be rather loosely connected to each other, this impression is only relative, being caused by our focus on a too low level: When seen in a more coarse scale, for example if the time intervals become longer, the loose connections become tight again, and another higher-level mixture model can be constructed. This way, a self-similar structure of mixture models is created.

It seems that a special conceptual framework would be needed here to master the organisational complexity between the mixture models. See Fig. 7: Assume that each block represents some subsystem, or an individual mixture model. Seen from outside, a block represents the essence of the corresponding system, the evaluation module extracting the relevant qualities to be seen by the other blocks. Note that this structure is well compatible with Fig. 3, the evaluation of data just being distributed in appropriate blocks. To implement an individual mixture model, an "exclusive OR" type nonlinearity is needed on top of linear feature structures to select between candidate Gaussians, applying the best-match criterion (21) in each submodel; the underlying functionality (matching of linear features) remains the same. This way, the mixture models can efficiently be implemented, and the known a priori system structure can be imposed. All submodels, no matter if they represent individual devices, larger processes, or entire plants, can be written using such blocks, so that there will be a hierarchy of structurally identical model constructs.

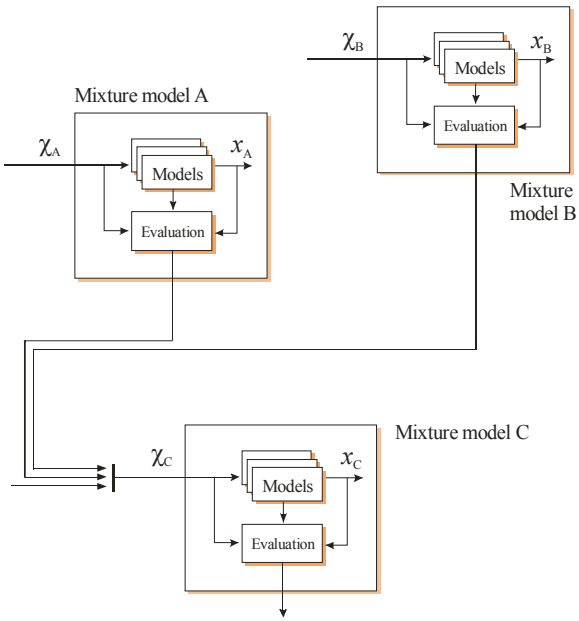

Figure 7. Generic structure among subprocesses

THEORETICAL ISSUES

No matter whether all of the above discussions really reveal something essential about the future prospects or not, there are necessarily some underlying principles that will be shared by all future environments – they will most certainly be more data-oriented than the today's tools. Generally, what is possible and what is not in such environments – something *can* be said about that. The ultimate goal would be an *autonomous modelling system*, being capable of appropriately adapting to changing environments. Even if this goal may be too ambitious, analyses may help in redirecting the efforts in the most promising directions.

Deep Questions

In complex systems research, it is assumed that interesting patterns emerge from massive iterations. When is this possible? And, what is *interesting*, to begin with? To answer such questions, one needs to study truly deep philosophical questions. The keywords here are *epistemology*, *ontology*, and *semantics*.

Epistemology is a field of philosophy that studies *what one can know*. Traditionally, the basic concepts in this field involve the notion of *truth*. However, truth is a very idealistic concept; it is clear that no methodology that is based on data and measurements, can ever have anything to say about what the underlying "real world" is. Not to get stuck in such basic discussions, a simplification is possible – rather than speaking of truth, one can concentrate on *relevance*. Something is relevant if its effects can be seen in the data; if something is invisible it can be forgotten (even though it is not necessarily *false*). Only quantities that are measured can be included in the model; the sensors determine what the world is like. In this sense, the approach is extremely *empiristic*. And, as David Hume has said, in such models causality cannot exist – causal relationships cannot be seen in data, only correlations. This means that the models can only capture static patterns, if no a priori causal structure is explicitly defined.

Ontology studies what kind of objects there really exist in the world. The above view of epistemological issues makes this issue also easier: The phenomena that are visible in statistical data are relevant, they *exist*, and they will be manifested in models, determining the *beliefs* of the model. Of course, it is extremely dangerous to believe in the model more than in reality – but assuming that our perception mechanisms follow the same principles when modelling the world (see later), the model may still appropriately reflect our *subjective* world!

To have something meaningful, non-trivial automatically emerge out from some mindless mechanical machinery, the domain-specific meaning, or *semantics*, must *somehow* be captured in the functions. Capturing semantics is an age-old problem in philosophy, and it has been extensively studied there – these discussions have been exploited in artificial intelligence research, but the results have been rather *shallow*. The domain-area expertise and understanding is traditionally hand-coded into the model in symbolic form. However, if the models remain purely on the syntactic level, no *real* semantics can be buried in the data structures. If the model is based merely on connections between other data structures, so that symbols determine other symbols, the *grounding of semantics* remains *floating*, and the problem of *infinite recess* (dead end of *hermeneutics*) cannot be avoided – as observed by philosophers.

It seems that some level of *artificial understanding* still *can* be implemented. Rather than speaking of real semantics of a system, one can speak of its behaviour in an environment and its reactions. In this way (again, using terms adopted from philosophy) the grounding of semantics is *naturalistic* (reactions to inputs) and *contextual* (reactions to neighbours). And it is *simulation* that makes it possible to reach automatic "understanding": The simulations supply for the grounding, hard data, and the evaluation (as shown in Fig. 3) supplies for interpretation and expert understanding. This means that the proposed structure can carry some "formalised semantics", and emergence of something interesting *may be* possible.

Connections to Cognitive Science

Does there exist an objective world outside our subjective sensations? This age-old problem was practically solved by Immanuel Kant: Even though our observations are personal, the mechanisms that process them are the same to all, meaning that the same kinds of data structures are constructed by everybody in the same circumstances. Here it must be assumed that there exist some underlying relationships between observation units, so that the existing mechanisms can detect them.

The automatic modelling tool is facing the same kinds of challenges. In both cases, there must exist some statistical relationships among the data units if structure is to be determined. As was motivated, mixture models with data clusters and their substructures seem to offer a nice framework for industrial systems – how about more general systems? Would the same framework be motivated? And if so, even more interesting questions emerge: If the mental machinery has adapted to optimally capture the statistical properties in the observation data, there must exist some connections between the optimized data structures and mental representations. Modelling of a large system is like modelling the world itself.

And, indeed, there are connections – one just needs to slightly change the labels. Traditionally in cognitive science, the basic elements when structuring the world have been *categories* and their *attributes*. The categories can be identified with data clusters (centers of relevant data) and the attributes with the degrees of freedom within the cluster (typical variations around the nominal prototype) – for example, see [8].

Structure cannot be deduced when only data is seen. However, because the clusters/categories determine the structural entities, within the selected framework structure can be learned from data, at least in a narrow sense. If the underlying essence of symbols is in the form of data distributions, the painstaking problems of symbol grounding become less acute. However, the problem of causalities cannot be solved – seeing only the data, one cannot say whether the relationships between data units are only correlations, or if there exists a real dependency between them.

New Concepts and Languages

The role of languages as skeletons beyond our thinking (as Ludwig Wittgenstein put it in his "Tractatus") cannot be overestimated. One simply cannot think of something if there are no appropriate concepts and frameworks for expressing one's thoughts. A "New Science" necessarily introduces new ways of thinking, and it is not only new tools, but a deeper shift in ways of thinking that will be needed. There will be not only new concepts, but the whole framework changes – new interpretations that would have been meaningless become basic constructs for structuring the world.

To reach models having closer connection to real systems, some kind of data-centered approaches are needed; however, numeric entities are non-symbolic, and there is no way to express them in any natural language. It is *mathematics*, and specially linear algebra and matrix calculus, that is the "compiler" for such data-oriented formalisms[1]. This is due to the fact that objects implemented in this framework have a dual interpretation also as data distributions, and functionalisation of the descriptions is based on the statistical structures and relationships. Executing a program becomes a pattern matching process rather than direct sequential computation; computation that is based on matrix calculus makes parallel execution of matching processes simple. Recall of variables concerning the system state is based on "associative regression" rather than functional manipulations.

Looking at the features-based models, there are some consequences what comes to implementing practical programming languages – a formalism reflecting this underlying realm will be very different from those that one is familiar with. First, now the category (cluster) centers and the attributes (subspace axes) have equal representation; classes and subclasses, objects and their properties have the same real-valued vector form. There needs be no distinction between classes and their properties: The feature-based attributes are implemented as other categories, meaning that constructs are used to determine each other. This offers a key towards *generalised hierarchies*: Variable values (preliminary analyses of the input) are reiterated, as the features refer to each other. Indeed, rather than speaking of hierarchies, one should speak of *networks*.

The second property of the features-based language is that rather than being symbolic and "crisp", references to other constructs are continuous-valued. The links between units are quantitative rather than qualitative – connections are not "yes" or "no", but, indeed, the *fuzzy subclass* structure is spanned (see [17]). This means that it is not only the subclass that inherits properties from its superclass, but the superclass properties are also affected, to a lesser extent, when new subclasses are introduced. Multiple inheritance can also be implemented in this framework in a natural way[2]. The numeric, redundant nature of representations also means increased robustness, and *graceful degradation* rather than abrupt collapse in those cases where the expressional power of the model is exceeded.

If a concrete programming language were to be designed based on the above principles, one can note that there would be a close relationship to today's object-oriented formalisms. The clusters determine the classes and the degrees of freedom within them determine the (numeric) properties, as given by elements in matrix C. The classes can also contain reasoning subprocesses, or algorithmic, sequential computation [11,12]. If the "Alternative Approach", as presented above, is implemented, the internal structure of the class becomes, of course, more sophisticated, containing features determining the subdistributions, and a mechanism for determining the outputs (or "evaluation").

---

[1] Thinking pragmatically, the emphasis on the role of mathematics is, actually, the Achilles' heel of approaches of the proposed type. Remember what happened with *modern control theory* with its grand elegance and power. Because of the mathematically sophisticated structures and tools, it seems, it never became widely used in industry – the operators are either using the age-old PID, or are clamouring for "postmodern" soft-computing methods, where "mathematics (or models) is not needed". One should always take into account the strange dynamics of the complex systems involving humans!

[2] All these ideas concerning classes and their properties are, in fact, well in line with observations concerning the human categorization process – the Aristotelian taxonomies, and the object-oriented data models as their direct implementations, are not well suited for real-life modelling tasks

Contribution to Automation Engineering

The title promised that in this paper industrial automation will be discussed, but, one could protest that little has been said about it up to now. The above discussions concentrated on modelling, or abstracting of processes; however, it can be claimed that the role of models is becoming more and more important in practical engineering work.

When comparing the new modelling approaches to the old ones, it seems that the main difference is close connection to data. When fieldbuses, etc., are coming to factory floor-level, *availability* of the processes is enhanced. New functionalities (condition monitoring, etc.) are being included in the devices, and new kinds of measurements (image analysis, etc.) are being installed, so that more and more data is delivered. The models need to support the instruments that are getting "smarter", and the intelligent field devices should be integrated within the modelling system ("plug-and-play"), delivering up-to-date information of the device functioning. If the modelling system can utilise the measurements in a flexible way – as hypothesised above – the model integrity and consistency can automatically be maintained even though the system properties gradually change (support for "hardware-in-a-loop"). The model constructs can be adapted according to the observed data, so that the features can change more or less automatically to better match the real behaviour. As the interfaces between the model levels become homogeneous, one can get nearer to the agent paradigm: Independent actors can be integrated in the model as long as they deliver the data promptly. If the interfaces can be standardised, major part of the modelling task can be distributed to the device manufacturers and vendors, and a new level of model accuracy can be reached.

When larger units of the process plant are provided with homogeneous, data-based models, different kinds of exploratory data analysis methods can be applied to the observations, and the process data analysis can result in deeper understanding of the relevant phenomena and dependencies taking place in the system, facilitating (multiobjective) optimisation, including robustness considerations, or perhaps even automatic (self)organisation of the process structures and parameters. Indeed, the close connection between the model and process data makes it possible to integrate the model and the process. As shown in [12], observations from the real system can be applied to tune the model parameters – but, as also shown in [12], the system parameters (controller settings) can similarly be adapted to make the system behaviour look more like the model behaviour. This way, some kind of *model reference control* can be implemented.

Many standard practices may be changing. As shown above, new and more robust parameter estimation schemes become available; what is theoretically more interesting is that new views of *adaptive control* are opened. When looking at the system hierarchy, it can be seen that the signals themselves are operated on the lower level, whereas the parameters reside on the higher level. The traditional problems of adaptivity caused by the bilinear nature of the models (variables and parameters simultaneously changing) are avoided because the dynamics is divided on the two separate levels, both of which can be linear. Indeed, such discussions may lead to a higher-level theory for *gain scheduling* and *multi-model control* [8,10].

CONCLUSION

It is difficult to get a grip of large-scale systems. One should master such plants, yes, but what does that *mastering* really mean? It can be claimed that the key point in mastering a complex large-scale system is to understand what is happening, what is relevant and what is not in the wealth of data flow. In truly complex systems (in economical or social systems) the dynamics follows its own laws, and one cannot affect it by simply exerting brute force. In such cases it would be enough if one could be aware of what will be happening and prepare for the consequences. The more complex the system is, the more important it is just to see the *state* of the process. A good model should capture the relevant phenomena, and, hopefully, also reveal what should be done to reach a better state.

No matter how much the tools develop, it will always be the human who takes the final responsibility. But the expert's task should be simplified: Clever automatic analysis of the available information is necessary. It is not only simple filtering of data that suffices, but something *cleverer* is needed. The modelling tools should be capable of imposing structure onto the data. In this sense, feature-based models – like the proposition above – seem to be most potential in their extreme simplicity.

The proponents of complex systems are speaking of this field as a Theory of Everything. There is the "big hammer" syndrome: Everything is seen from a rather narrow perspective. It turns out that *semantics* is very domain-specific; approaching the domain fields from above, using too universal tools, lacks concreteness and content. Starting from below, from practical application fields, it may be that, paradoxically, wider perspectives can be seen. Above, we started with analysis of one complex system, an industrial automation environment – but it turned out that we could proceed towards added generality. As shown above, it may be possible that there exist fundamental links between very different complex system domains, industrial automation and cognitive processes. One thing is certain: There are interesting times ahead, no matter what the new approaches and tools finally turn out to be like.

ACKNOWLEDGEMENT

During this work, the author received financial support from Nokia Research Center (Helsinki, Finland); this support is gratefully acknowledged.

REFERENCES

[1] A.-L. Barabasi, 2002. *Linked: The New Science of Networks*. Perseus Books, Cambridge, Massachusetts.

[2] A. Basilevsky, 1994. *Statistical Factor Analysis and Related Methods*. John Wiley & Sons, New York.

[3] L. von Bertalanffy, 1976 (revised edition). *General System Theory: Foundations, Development, Applications*. George Braziller, New York.

[4] A. Bunde and S. Havlin (eds.), 1994. *Fractals in Science*. Springer-Verlag, Berlin.

[5] J.M. Carlson and J. Doyle, 1999. HOT: A mechanism for power laws in designed systems. *Phys. Rev.*, E **60**, pp. 1412–1427.

[6] J. Horgan, 1997. *The End of Science: Facing the Limits of Knowledge in the Twilight of the Scientific Age*. Helix Books, New York.

[7] A. Hyvärinen, J. Karhunen, and E. Oja, 2001. *Independent Component Analysis*. John Wiley, New York.

[8] H. Hyötyniemi, 1998. On mental images and "computational semantics". *Proceedings of the 8th Finnish Artificial Intelligence Conference STeP'98*, Jyväskylä, Finland, September 7–9, 1998, pp. 199–208. (*http://www.control.hut.fi/hyotyniemi/publications*)

[9] H. Hyötyniemi, 1999. From intelligent models to smart ones. *Proceedings of IPMM'99 – The Second International Conference on Intelligent Processing and Manufacturing of Materials*, Honolulu, Hawaii, July 10–15, 1999, Vol. 1, pp. 179–184. (*http://www.control.hut.fi/hyotyniemi/publications*)

[10] H. Hyötyniemi, K. Koskinen, and K. Saloheimo, 2000. Calibration of an X-ray fluorescence analyzer using clustered components. Proceedings *of the IFAC Workshop on Future Trends in Automation in Mineral and Metal Processing*, August 22–24, 2000, pp. 219–224. (*http://www.control.hut.fi/hyotyniemi/publications*)

[11] H. Hyötyniemi, 2001. Multivariate regression – From static models to subspace identification. *Proceedings of IPMM'01 – The Third International Conference on Intelligent Processing and Manufacturing of Materials*, Vancouver, Canada, July 29 – August 3, 2001 (CD-ROM). (*http://www.control.hut.fi/hyotyniemi/publications*)

[12] H. Hyötyniemi, 2002. Towards new languages for systems modelling, and On emergent models and optimization of parameters. *Proceedings SIMS 2002 – The 43rd Conference on Simulation and Modelling*, Oulu, Finland, 26–27 September, 2002, pp. 77–82 and 45–50. (*http://www.control.hut.fi/hyotyniemi/publications*)

[13] H. Hyötyniemi, 2002. Studies on emergence and cognition – Parts 1 and 2: Low-level functions, and High-level functionalities. *Proceedings of the 10th Finnish Artificial Intelligence Conference STeP'02*, Oulu, Finland, 15–17 December 2002, pp. 286–299 and 300–312. (*http://www.control.hut.fi/hyotyniemi/publications*)

[14] H. Hyötyniemi, 2002. Life-like control. *Proceedings of the 10th Finnish Artificial Intelligence Conference STeP'02*, Oulu, Finland, 15–17 December 2002, pp. 124–139. (*http://www.control.hut.fi/hyotyniemi/publications*)

[15] H. Hyötyniemi, 2002. Towards Perception Hierarchies. *Proceedings of the 10th Finnish AI Conference STeP'02*, Oulu, Finland, 15–17 December 2002, pp. 111–123. (*http://www.control.hut.fi/hyotyniemi/publications*)

[16] H. Hyötyniemi, 2002. On *the universality and undecidability in dynamic systems*. Helsinki University of Technology, Control Engineering Laboratory, Report 133. (*http://www.control.hut.fi/hyotyniemi/publications*)

[17] B. Kosko, 1993. *Fuzzy Thinking – The New Science of Fuzzy Logic*. Hyperion, New York.

[18] A. J. Niemi, H. Hyötyniemi, and R. Ylinen, 1999. Image analysis and vision systems for processing plants. *Proceedings of IPMM'99 – The Second International Conference on Intelligent Processing and Manufacturing of Materials*, Honolulu, Hawaii, July 10-15, 1999, Vol. 1, pp. 11–20. (*http://www.control.hut.fi/hyotyniemi/publications*)

[19] S. Russell and P. Norvig, 1995. *Artificial Intelligence – A Modern Approach*. Prentice Hall International, Englewood Cliffs, New Jersey.

[20] H.A. Simon, 1996 (third edition). *Sciences of the Artificial*. MIT Press, Cambridge, Massachusetts.

[21] S. Wolfram, 2002. *A New Kind of Science*. Wolfram Media, Champaign, Illinois.