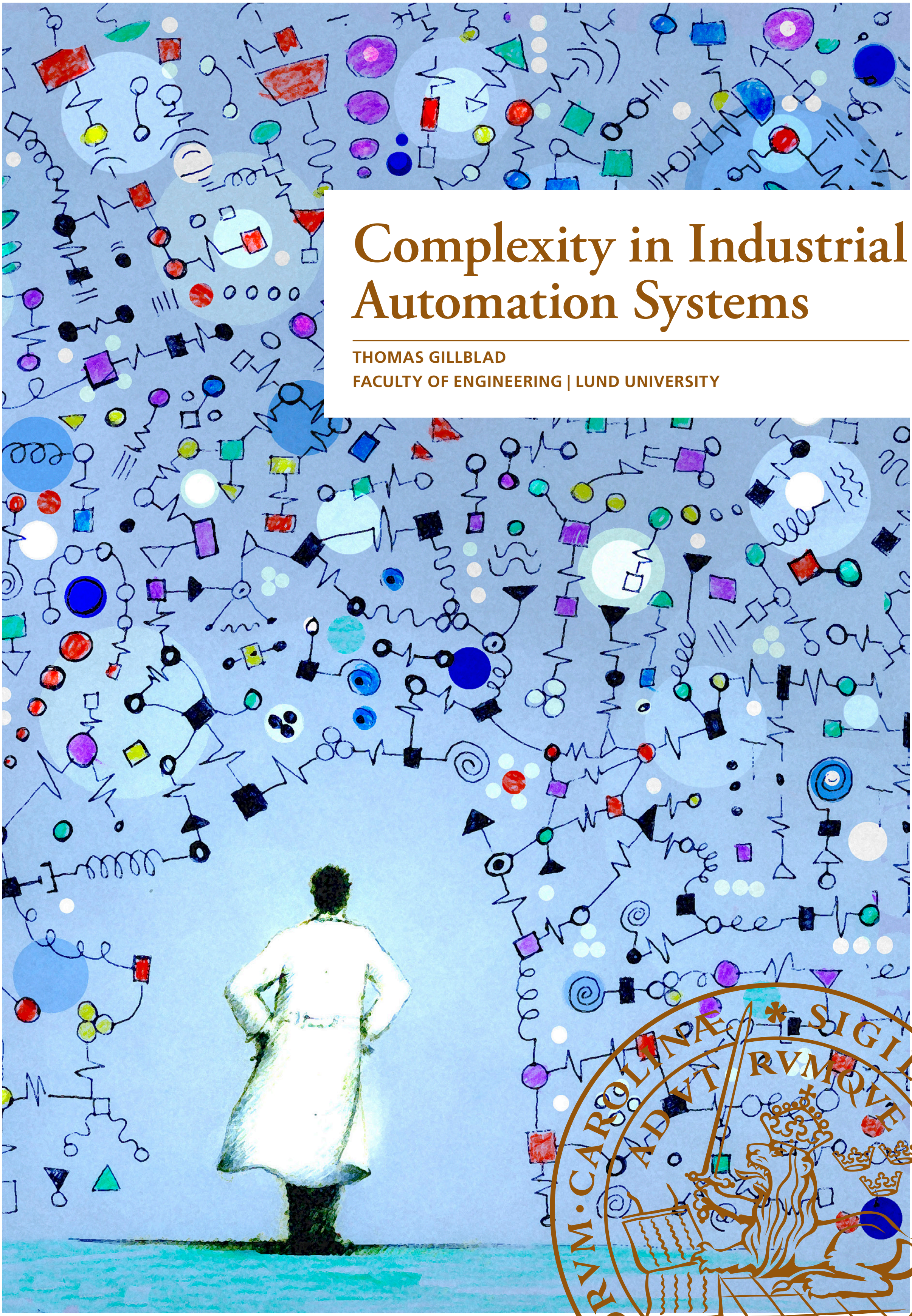

Complexity in Industrial Automation Systems

THOMAS GILLBLAD
FACULTY OF ENGINEERING | LUND UNIVERSITY

Thomas Gillblad is currently working as an entrepreneur.

He has been enrolled at Lund University at the division of Industrial Electrical Engineering and Automation (IEA) for his Ph.D. studies. The area of research is complexity in automation. He has a B.Sc. majoring in Physics and a M.Sc. degree in Engineering Physics from Uppsala University. Active 11 years at Texas Instruments, working with petrochemicals and as a member of the International Organization for Standardization (ISO). He has started half a dozen companies, one of which was merged and introduced on the stock market. The current activity involves using findings from this thesis to improve control applications.

Printed by Media-Tryck, Lund 2020 NORDIC SWAN ECOLABEL 3041 0903

LUND UNIVERSITY

Lund University
Faculty of Engineering
Department of Biomedical Engineering
Division of Industrial Electrical Engineering and Automation
ISBN 978-91-88934-92-5
CODEN:LUTEDX/(TEIE-1089)/1-112/(2020)

9 789188 934925

Complexity in Industrial Automation Systems

Thomas Gillblad

LUND UNIVERSITY

LICENTIATE THESIS
by due permission of the Faculty of Engineering, Lund University, Sweden.

Organization LUND UNIVERSITY	Document name LICENTIATE THESIS
	Date of issue Mar 18, 2020
Thomas Gillblad	Gillblad Technology Investment AB

Complexity in Industrial Automation Systems

Abstract

As automation systems grows in number and scope, in particular as a result of systems being connected to and interacting with each other, the number of situations increases where various systems may perform in ways that were not intended. This situation is considered being mainly the result of increased *complexity*.

This thesis studies the concept of *complexity* from a technical perspective and aims at classifying the types of complexity present in industry today. It also proposes an approach for handling the design of automation systems with inherent complexity.

The method chosen is to present a hypothesis and try to validate it by applying the principles to a number of manufacturing processes that have been identified by the author from personal experience of automating the processes.

The research method is to study over two dozen automation systems, analyze their characteristics and group them into distinct complexity categories. These are categorized using their boundary conditions, and a new concept, the *function state matrix*, is introduced to describe the limits of the respective processes.

Finally, a method in proposed how to analyze the control authority of different control systems in order to improve the areas of control. One system is used as a validation example in an appendix and other areas are suggested for future work.

Key words

Complexity, Automation, Manufacturing Process Modelling, Functional Complexity, Operational Complexity, Integrative Complexity, Recursive Complexity, Control System Examples, Function State Matrix

Classification system and/or index terms (if any)		
Supplementary bibliographical information		Language English
ISSN and key title		ISBN 978-91-88934-92-5 (print) ISBN 978-91-88934-93-2 (pdf)
Recipient's notes	Number of pages 112	Price
	Security classification	

Signature

Date Mar 18, 2020

Complexity in Industrial Automation Systems

Thomas Gillblad

LUND UNIVERSITY

Cover illustration front: Drawing "Scientist standing at opening of complex lines and shapes". Artist: Gary Waters. Copyright: Illustration Source. Printed with permission.

Cover illustration back: Author’s portrait

Faculty of Engineering
Div. of Industrial Electrical Engineering and Automation (IEA)
Dept. of Biomedical Engineering (BME)

CODEN:LUTEDX/(TEIE-1089)/1-112/(2020)

ISBN 978-91-88934-92-5 (Print)

ISBN 978-91-88934-93-2 (PDF)

Printed in Sweden by Media-Tryck, Lund University
Lund 2020

Printed matter 3041 0903

Media-Tryck is a Nordic Swan Ecolabel certified provider of printed material. Read more about our environmental work at www.mediatryck.lu.se

MADE IN SWEDEN

Dedication

I dedicate this thesis to my wife Åsa, who has provided constant encouragement, without which it would not have been completed.

Content

Acknowledgements

This thesis is very much the result of an idea originating from Professor emeritus Gustaf Olsson, who came up with the suggestion that I should use the information in my notes from automation processes that I had been working with as a base for my thesis.

The path to find useful synthesis information and to draw conclusions from them has been very time consuming and the difficulty in extracting useful data has been very taxing and sometimes required encouragement. This has been amply provided by Associate Professor Ulf Jeppsson, who patiently and efficiently checked all results and provided valuable contributions.

I also want to thank Dr. Gunnar Lindstedt, who complemented my knowledge in areas of high practical value.

Professor emeritus Gustaf Olsson has also been a constant source of encouragement and has provided valuable insights as to which parts of the synthesis that have the most interest.

List of figures

Acronyms

API	American Petroleum Institute
CPSOS	Cyber-Physical System Of Systems
DC	Direct Current
ERP	Enterprise Resource Planning
MTBF	Mean Time Between Failure
IEEE	Institute of Electrical and Electronics Engineers
MES	Manufacturing Execution Systems
MID	Molded Interconnected Devices
OPC	Open Process Connect
OS	Operating System
IoT	Internet-of-Things
IIoT	Industrial Internet-of-Things
RISC	Reduced Instruction Set Computer
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Universal Markup Language
VTQ	Value Time Quality
XML	eXtended Markup Language

1. Introduction

The material for this thesis is based on personal experience, gathered during a long career in delivering industrial control systems. The author has been involved in the design of many such systems, and much of the material is based on notes from projects that have been realized during his career. The application sites from where the references are drawn are listed under application references.

1.1 Purpose and limitations of the thesis

The purpose of the thesis is to draw conclusions about how complexity appears in industrial automation systems. The basis for this thesis is practical experience obtained while implementing various automation systems in different application areas. In each case, implementation difficulties appear. This thesis argues that there are only a limited number of principally different types or categories of such implementations, which combined together with their respective limitations (boundary conditions) form the topic of ***complexity***.

It is the aim of this thesis to try to devise a workable approach in order to create practical methods to handle technical complexity in automation systems. However, complexity is handled as ***technical complexity***. Complexity caused by interaction between humans and between human and machine is not specifically covered.

1.2 Challenge to handle complexity

There are several challenges to the topic of complexity. The major one is that the word complexity means different things to different people. In fact, it has been stated that "even among scientists, there is no unique definition of complexity – and the scientific notion has been conveyed using particular examples ….". This has been stated by Johnson (2009), who adopts the definition of "complexity science" as "the study of phenomena which emerge from a collection of interacting objects". In this context, this thesis treats complexity as the opposite of simplicity. It is easier to define simplicity, for example as processes that have one hundred percent complete models, even if such models, and combinations thereof, can appear complex. A complete model is here defined as a model that has all its states defined and all relations between inputs and outputs in the model are known. In this respect, complexity is more related to difficulty, and this thesis will use experience from various processes to exemplify complexity by relating to which part of the respective process is most difficult to control. The approach of using examples is in the spirit of the idea by Johnson (2009).

Complexity in the context of this, is not directly related to the fact that some manufacturing processes can be considered *complicated*. For example, the manufacturing of a commercial airliner is complicated but it is not complex from a control standpoint.

This thesis tries to define complexity related to industrial automation. Another characteristic is that the thesis deals with synthesis, whereas most of the academic research work in this area is concerned with analysis, mainly of the behaviour of various manufacturing processes. A major obstacle is validation of findings. This research argues that complexity is closely coupled to control authority and its consequential boundary conditions, and almost every process owner is unwilling to drive processes to their corresponding limits only to prove an academic case. Simulations are an alternative to this, but in many cases simulation will not be able to simulate real life. In some cases simulations can have difficulty in encompassing process variations. Simulations have been made for technical reasons in the construction of airplanes and submarines, but are more rare when it comes to manufacturing processes. Here we argue that a lot of complexity arises from the fact that rare occurrences are not handled properly/foreseen in most control implementations. Also, most systems can handle one contingency, sometimes two but almost never three or more.

There are several examples of studies of the general aspects of complexity. One recent book by West (2017) gives insights in the general nature of complexity.

Furthermore, there are studies on the limits of our potential knowledge, e.g. some systems may be so complex that they test our capacity of understanding. In this respect, industrial models become influenced by ontological questions on how to use available information and epistemological questions about which areas we should investigate. Ontology is the study of being and epistemology the branch of philosophy that concerns itself with the study of knowledge. One such study is the book by Yanovski (2018). In Santa Fe in New Mexico, USA, there is a whole organization dedicated to the study of complexity.

The results presented are of a qualitative nature, since data on complexity are hard to measure. Instead, the qualitative approach has been to find useful patterns, in the spirit of Henri Poincaré (1902).

The findings in this research are presented in the academic context pioneered by Karl Popper (1959), where conclusions are presented and illustrated by examples. All processes analyzed have been found to belong to one of the four categories presented in Chapter 6.

1.3 Approach to define complexity

Complexity appears when the automation systems are not able to handle all states pertinent to the control authority.

At first, this definition may seem a bit out of place, surely a control system must be able to cope?

Alas, this is not the case in the majority of cases. Almost everything is designed with preconceived limits to its operation, and in most cases it is easy to envisage a situation outside the "normal boundaries" of activity. It is simply too expensive to build systems that can handle every situation, and the fact that the systems will not operate predictably outside the control authority brings uncertainty, which is a contributor to complexity. A simple example may illustrate the idea:

Consider a common hand maneuvered faucet. It can open and close the supply of water to a sink. It will work in every case, and the sink can normally accommodate even a full flow from the faucet.

This device does not contain any complexity: Even in a big building with a thousand faucets and a thousand sinks, the combined system is still not complex since the water evacuation systems can normally handle the flows from every faucet. The water supply system may not be able to provide full flow in every unit, but this will ***still*** not create complexity. The system will be able to handle all combinations of faucet flow settings.

In many hotels and airports automatic faucets are present, which have sensors that open the flow in the faucet when a hand is placed under it. This is a simple automation system. However, the automation of this process does not add any complexity. Even with large configurations of faucets the combined system will still ***not contain any complexity***.

Another device in the vicinity is the flushing toilet. It has a tank and an on/off valve, which releases the water and then refills. A single toilet of this type does not contain any complexity. However, if there are many toilets in an apartment building and all of them are flushed almost simultaneously, chances are that the toilets on the lower floors will overflow. The reason is that the water evacuation system in most cases is not dimensioned to handle this improbable event. The combined system therefore contains complexity as defined in this thesis, because there are system states outside the boundary conditions where the controls fail to handle the desired output. The water is supposed to be evacuated, but it goes in the wrong direction in rare cases.

1.4 Thesis overview

The thesis is organized in chapters. Firstly, the origins of complexity are listed (Chapter 2), then the causes of automation complexity are listed in Chapters 3 and 4. A number of principally different processes are analyzed (Chapter 5), and an approach to categorize the different processes according to their complexity types is made (Chapters 6-8). Finally, conclusions and suggestions for future studies are made in Chapters 9 and 10. A sample application of the methods proposed is presented in an Appendix.

Since many manufacturing processes can be considered to encompass more than one type of complexity, the method chosen has been to organize them according to main characteristics. When developing a new manufacturing process, there is frequently one or a few parts of it that are the most difficult ones to master. This is then used to define the ***main*** characteristic, since this in many cases coincides with the area of highest complexity. Over time, automatic manufacturing techniques has advanced significantly and as a result, a process that previously was unautomated through lack of development or technically out of reach has reached a point where it has become manageable from a control standpoint.

This thesis is limited to complexity found in industrial automation. A comprehensive description of the application area is to be found in the textbook by Olsson and Rosen (2005).

1.5 Main findings

The main finding is that complexity of various systems can be analyzed using a single method and that the control systems constructed based on this analysis can be categorized into a limited number of structures. Validation of findings have been done on a few simple networked systems.

In principle, the methods can be used to calculate:

- when a big system comprised out of many subsystems becomes so large that the resulting projected reliability of ***the whole system*** will be less than some threshold value,
- the number of sensors required to validate a dynamic model,
- the minimum number of functions that have to be implemented in a system in order to create/establish a space of control authority.

The space within which the control authority exists is affected by boundary conditions. If too many constraints are applied to a single system it may happen that the resulting area of complete control is zero. This means that the control system always will have at least one variable stuck to a boundary condition/value. However, careful analysis of complexity can result in an increase in the probability that a certain process can stay within its space of control authority. One such example is given in the Appendix.

In fact, this may be the most applicable result of this thesis. By applying the methods proposed, an automation system can be implemented in order to obtain a certain functionality level. This means that it will be designed to cope with a level of complexity, and the control strategy can be augmented using the boundary conditions so that the setpoints are dynamically reprogrammed to keep the process within the control authority of the actuators. One method is to design the dead-bands around each variable so that the process remains controllable, albeit this can reduce the precision in output.

Also, the size and the corresponding complexity of a large system may be so great that standard sensors cannot be used, since their individual reliability is limited. Therefore sensors with higher reliability, such as sensors with in-built redundant functionality are needed.

2. Origins of complexity

Complexity originates from several sources, depending on the process we are trying to control.

(i) The most common case is where the complexity is ***inherent***, by the fact that the relationships between the inputs and the outputs are not fully known or there are too many (or too few) known parameters that can affect the output. Typical systems would be various biological processes, but it could also be processes that are so fast that the control cannot act before the systems move to another state.

(ii) Time varying systems. Another source of complexity is that the conditions the system is set to operate under, change over time. Even if we tune our algorithms the system will act incorrectly because the gain or other parameters become inappropriate for the new set of conditions.

(iii) Instability caused by too many interconnections. A third and more insidious source is when systems become unstable because more and more subsystems become connected to it and the combined effects are not correctly envisaged when the designer made the model which the system programming was based on.

(iv) From corruption of outputs. Another but rarer type of complexity occurs when a control loop has limits on how many iterations the loop can perform before the result becomes corrupt. It is well known that digital algorithms will accumulate rounding errors but there are real life processes, for example recycling, where an infinite loop is not technically feasible or even possible. Entropy is an example of this. The entropy of modelling has been studied at Linköping University by Krus (2018).

(v) As a design consequence. In many processes the designer strives to reduce variations in a chosen parameter. However, this can result in an increase in the variability of other parameters. For example, if a road speed bump is introduced to reduce variation in vehicle speed, this can increase emissions from engines and increase wear on suspension systems. Many similar cases can be found via manufacturing execution systems, supervising production. If the operators try to narrow the band within which a parameter is allowed to operate, it can lead to greater variation of other parameters.

2.1 Main process categories

In order to analyze the various cases, one must start to investigate which are the main characteristics of the process. Fundamentally, the control application area can be divided into two main categories:

1. Process industries, where the product produced is a consequence of the process parameters. Once produced, the resulting product is classified to a ***grade***, and then sold depending on the value or characteristics of the grade.
2. Manufacturing industries, where the end product is either acceptable or discarded. This is according to a binary 0/1 scale where 1 is passable and 0 is not accepted.

Hybrid industries are treated as combinations of the two above.

This division is really the same as a division among analog and digital functions. The first one allows an infinite number of solutions whereas the other has a finite number of solutions. The design will be a result depending on what the production goal is.

The fact that a process is complex does not mean that it is ***uncontrollable.*** Most control systems are designed in such a way that if they reach areas outside their control authority, they will do nothing. One famous example is the hype around the year 2000 ("the millennium bug"). Only a few examples of uncontrolled behaviour were recorded. Worse cases are when active control is necessary to maintain stability. Important examples are found in safety systems in nuclear power plants. The Fukushima disaster in 2011 in Japan is a notable example when absence of active control renders parts of the process uncontrollable. Therefore, in certain areas, construction of systems that will not remain safe in the absence of power is very unadvisable, such as systems that will burst at sub-zero temperatures due to water-to-ice expansion properties. It should be emphasized that most complex industries are hybrid in nature.

In a process industry, not all control problems are continuous, but have to be complemented with discrete (event driven) decisions. Also, most discrete industry applications also include continuous control systems, like velocity and position control systems in machines.

However, true continuous control is rare in real life. Most automated processes are discontinuous.

2.2 Production goals

The reason for considering production goals is that they set the framework for the basic mathematics of the automation process. The fundamental equations describe inputs, which are acted upon by a function that generates the outputs. The production goals set the basic parameters of inputs and outputs.

In a process industry, the goal is frequently to provide the maximum of usable finished products from a given set of raw materials. The efficiency of the production process is often referred to as ***yield***.

Example 2.1 *The petroleum refining industry*

> A common and early example of a process industry is petroleum refining. Basically, a refining industry is divided into extraction of raw material, called ***upstream*****,** and raw crude processing, which is further divided into ***main process*** areas and ***off-sites***. The finished products are then distributed to consumption, which is called ***downstream***. This process will be discussed in more detail in Chapter 5 so we summarize somewhat here. The main process is the core and the production goal is to adapt the process so that the yield generates as much as possible of the products that can be sold at a profitable price. However, the process has many aspects of complexity. Because it is analog in nature it is almost impossible to produce the desired products without producing some unwanted ones. Some unwanted products are truly useless, named ***slops***, and unfortunately the general procedure to dispose of them is to blend small amounts into good products. A handbook of the processes is available by Jones and Pujado (2006). The alternative would be a continuous accumulation. This problem is general with regard to process industries but more uncommon in the manufacturing industry (see further in Example 5.3).

In the manufacturing industry, the goal is to produce ***units***. The units have to meet certain criteria, mainly functionality to a specification. The goal is to produce all units in the production plan as efficiently as possible, using a minimum of resources during the manufacturing process and with as little waste as possible.

Much of the large manufacturing industries were developed in order to mass produce vehicles. A heavy duty transportation vehicle, a truck, consists of a load area, a frame, a power train and a cab (where the driver sits). In principle, sheets of metal from rolls are rolled out, stamped into pieces and welded together. This is a process which lends itself to automation because the requirements are very precise and variation is a minus. When people produce trucks, the production output quality varies with the skills of the workers. A computer system, however, can copy the actions of the best worker and the production goal is to achieve uniformity and minimum use of raw material and other resources. The process is divided into steps, where each step is given an ideal manufacturing time. This is called the ***tact***, and it determines the output that the manufacturing process is designed to achieve. In order for the unit to move from one step to the next, the unit is subject to tests to validate the result of the preceding production step, and this is normally a go/no-go decision. If the unit is approved, it moves on to the next step. If it is not, then it is taken off the line for rework or in rare cases, discarded as scrap. In the manufacturing industries, reclaiming of waste is often much easier than in the process industries as the units can be disassembled and sometimes recycled as used metal.

However, the general characteristic is that most decisions are digital in nature. There is seldom ambiguity over what an acceptable unit is and what is not. Here we will use the word ***compliant*** to denote what is acceptable output as a word covering both industry types.

3. Disturbances

Any production process is subject to disturbances. The most common cause of disturbances is non-compliancy to specifications. This is a major cause of complexity in production processes. Complexity arising from ***non-compliancy*** can be divided into:

- non-compliant raw materials,
- non-compliant production conditions,
- lack of resources (such as electrical power, steam, infrastructure, materials),
- in-process change of requirements,
- in-process disturbances due to process size (where the sum of rare occurrences affects the results),
- erroneous sensor values. Incorrect sensor values are a major source of disturbances. This topic is more elaborated upon in Chapter 3.2.

A non-compliant raw material situation can arise, for example, in a refinery cracker when the cracker has been fine tuned to achieve good yield from a well-defined feedstock and the producing unit receives feedstock with another quality. The purchasing department may have purchased crude on the spot market and the API gravity could be different, thus resulting in a different yield (the oil density is called API gravity by the oil industry).

3.1 External factor effects

Sometimes, factors that are outside the process design, such as environmental issues, can affect outputs. This is exemplified from various industrial production facilities. One case describes the manufacturing of electrical cables. These were cooled in a cooling tower after extrusion and the capacitance of the produced cables varied from summer to winter. It was found that the cooling tower had windows on the sunny side and when the sun shone through the windows, the ambient temperature in the production hall rose. This reduced the cooling of the cables and the wire coating layer of teflon became thinner.

Lack of resources is a frequent cause of complexity (and disturbances). An obvious case is when key individuals do not arrive at work as expected. One would think that this is not an automation problem, but the fully automated processes are still few and most production of goods contains some manual elements at some stage. Also, availability of key resources can be preferable when certain manufacturing operations are to be performed. At the Avesta steel mill, experienced operators were called in when a titanium slab should be processed. The author visited a factory in Kristianstad which had an old lathe that could mill stainless steel vessels with a 3 metre diameter. The lathe was manufactured in 1936, and when the factory owner was asked how long they would keep the machine the answer was: until N.N. retires. They had only one worker left who could operate the machine.

The effects of complexity on an organization has been studied at Volvo Cars by Fässberg et al. (2002).

A common cause of disturbance is when a machine placed as part of a production chain fails. Then all machines with a line dependence will be affected even if their respective functions are unimpaired. Another problem occurs if a number of machines are served by infrastructure systems with finite capacity. One example is a common weighing system feeding plastic pellets to the hoppers on a number of extruders. Some extruders may have to halt production if there is a temporary waiting list for new batches from the weigh-scale function. A frequent case of disturbance is limitations of electrical power, and this limitation may not even be of a physical nature. At the Sibes foundry in Gnosjö, which produces brass door handles, the production had to be adapted to the amount of hourly kilowatt hours contractually available, as exceeding this limit would have made the marginally produced quantity unprofitable.

The problem where production is dependent on a resource that defines the overall capacity controlled by MES (Manufacturing Execution Systems) is well studied. The investment necessary to increase overall manufacturing capacity by increasing capacity of the scarcest resource is called ***debottlenecking***. However, if a production system is limited by more than one bottleneck, then an investment to remove some of the bottlenecks will not be profitable until ***all*** relevant bottlenecks have been removed.

3.2 Effects of erroneous sensor values

Incorrect sensor values are a major source of disturbances. One obvious case is when incorrectly calibrated sensors yield erroneous values. However, there are many other sources, some of which are insidious in nature, and sometimes hard to detect. Examples of such cases are:

a. Step changes in measured value. This can occur when a primary sensor fails and a system switches to a backup sensor or if a filtered sensor signal suddenly changes its filter parameters.

b. Information “spawning”. When an A/D converter transforms an analog value to a digital signal, the resulting value is truncated. If this value is converted by two different signal processors and used for different purposes, one value transition can be treated as two or more events (see Example 4.2).

c. Data “shortage”. If a system is controlled by the help of a model, then a sufficient number of sensor/data points may be required in order to calculate correct values. If a sensor fails, most system revert to a preset value in the middle of the expected sensor range or freezes to the last accepted value. This will create a growing error until the sensor is working again. Another case of this effect occurs when sensors are taken out of service for calibration in continuous processes. The same effect can be observed when an automated process is shifted into manual mode.

Example 3.1 *A paint mixing plant*

> At a paint mixing plant supervised by a Microsoft Windows application, an automatic update during Saturday night caused the PC to post a question on Sunday to the morning operator if a USB driver software should be automatically updated. On Sunday morning no supervisor was available and the workers did not dare to reply to the computer generated question. The resulting effect was the loss of one full shift of production (8 hours).

In the process industry on-line changes of requirements is increasingly common. The most obvious area is production of electrical energy. Wind power is notoriously fickle but many manufacturing processes are also subject to sudden changes in demand, for example the electronics industries. Some products can be in great demand today but demand can fade quickly when a new product enters the same market segment. The waiting list for a Trabant car in the former East German Republic was several years but shrank into a few weeks in 1990 when cost effective alternatives (used cars from former West Germany) became available.

In the spring of 2015 elaborate plans had to be made for the German power grid because of a solar eclipse. A large portion of the solar energy supply disappeared from the grid during a few minutes.

A couple of examples will be presented in order to illustrate the type of disturbances caused by insufficient preparation for uncommon situations since the usual procedure is to consider the most common process states when designing systems but disregard the rare cases.

Example 3.2 *Three-way two-port valve*

The function state matrix (defined in Chapter 6) of a three-way two-port valve is in its easiest form very simple. It can be in any of three states: open left, open right or in transition. If one was to write a control program to maneuver the valve, the programmer may find that there is a 50% chance that the valve would be open to the right, a 49% likelihood to the left a 1% probability to be in transition at any given moment.

Figure 3.1 A three-way valve. From the publication American fire hose.

In reality, this is not the whole truth. The valve could be broken, the seal may be compromised so that it would leak a little in both directions, or it could be worn, so a transition would still occur but so slowly that the monitoring software would generate a noncompliance function alarm. More rare states could also be easily constructed. So, where is the problem? The problem lies in the fact that many control systems are only designed to handle the most common cases programmatically. In the rare cases, the actions of the control systems would be outside their local control authority and therefore a cause of complexity. In a small system, this is normally not a problem since a non-compliant state may never or rarely occur. However, if you have a large system with thousands of connected valves, like in a refinery? Then the sum of rare occurrences is no longer negligible.

The integrative complexity, further discussed in Chapter 6.3, caused by a large number of interconnected valves can be detected by talking to refinery operators: they are almost always adamant in getting equipment of the highest quality available. This is because they know from experience that this minimizes downtime resulting from sensor or actuator failures.

The valve model is treated in more detail in Example 7.2.

Example 3.3 *The PID controller*

> PID controllers come in different implementations, from simple PLC programs to advanced controllers with anti-windup and other features, which mainly are introduced in order to cope with out-of-boundary conditions. Large process industries, like a large refinery or a pulp and paper plant, contains numerous PID controllers but so does a big office building, for example for ventilation control. In the process industries with many controllers the gut feeling among the operators is conservative and they always want to buy the best equipment available on the market. Since the outputs are valuable and disturbances costly this makes sense. Furthermore, complexity analysis would confirm that this attitude is based on solid experience.

It is a common experience that climate control in buildings is not working to specification. Regulating a compressible medium like air with nonlinear actuators is not easy and one would argue that this application also needs qualified control. However, after having personally reset the integrative constant accumulators in building controllers after some inputs had been off line for some time, it has been noted that these large systems suffer from complexity effects due to system size.

This area needs more study, but based on experience from humidity control and air filter control in buildings, it is likely that the main cause of complexity in building control lies in the fact that controlling the properties of air is inherently complex. Air is a light, compressible medium and accurate flow meters are expensive, temperature measurements are often not representative and actuators nonlinear.

Another example of degraded control is a common heat pump. The unit converts electrical energy to heat by compressing and expanding a medium in accordance with a Carnot cycle. The Carnot cycle is a thermodynamic ideal cycle proposed by Sadi Carnot in 1824. It provides an upper limit on the efficiency that any thermodynamic engine can achieve during the conversion of heat into movement. The heat pump application has several boundary conditions. Firstly, the heat pump must not be larger than necessary because then the efficiency ratio between used and produced energy will be lower. Secondly, the unit should operate continuously since the process owner would not like to be without heat. The only stoppages should be during service/repair or if the unit has a major malfunction. However, heat demand can vary significantly over time and certainly be higher than the capacity of the unit. This means that the room temperature setpoint may not be met. If this would be handled through traditional PID control, then a wind-up error would accumulate, resulting in an overshoot of a room temperature when demand returned within the control authority area. Instead, the heat pump control algorithm will measure the time it is outside of the control area and lower the desired setpoint in steps until control is re-established. This is a procedure similar to what is used in a self-tuning regulator. Problems of this nature have been investigated by Abbate et al. (2010). This is a way of handling the inherent complexity in the application, but the user of the heat pump cannot set the temperature in a standard way, by only using for example the room temperature as setpoint. Instead the user is given a parameter between 0-10, which really is a measure on how hard the user wants the algorithm to try.

The conclusion from the disturbance examples is that this thesis argues that the area of desired control has to be analyzed *before* a control system is implemented. Consider designing a AC/DC converter from 240 VAC to, for example, 5 DC. If the DC requirement is 5,00 VDC, the system design has to be much more sophisticated.

4. Information fundamentals

Automated industrial production is fundamentally an information and feedback process. There is no basic difference when production is made by a person, e.g. manually, but we will not discuss those processes here since manual processes are more aptly handled by social sciences.

We will here consider:

- measurements,
- control signals,
- actuators.

When designing controls systems of individual processes there are various time constraints to consider as well as delays in data collection. When unit processes are combined new areas of concern occur, such as:

- homologation and synchronization of data,
- bottlenecks.

With homologation of data is here understood treatment of data so that they become comparable to each other, referring to the same time period, having sufficient accuracy, and generally of such a quality that the data can be used for calculation and control. From a complexity control standpoint, we are only considering bottlenecks that occur dynamically. Static bottlenecks are a process design criterion and will not be considered here.

The fundamental way to describe a control system is through a ***model.*** The model describes the relation between the inputs and the outputs of the system. The relationship between the inputs and the outputs is defined by the mathematical relationship that describes how variations in input values are transferred to changes in outputs. We will here refer to this as a ***function state matrix***. However, it is important to note, that this model is only used to describe/analyze the complexity of the control system. The function state matrix is linear, whereas many control system models are nonlinear. The concept is related to control theory models, because if it is multivariate and linear, it will become a transfer function matrix, known from control theory and in the single variable linear case it will be a transfer function. Traditionally, control theory will handle these relations and will divide controls into continuous and discontinuous processes. However, almost all man-made control systems are in reality discontinuous processes. All operate inside certain boundaries, which define the control authority. The function state matrix can be illustrated as a multi-dimensional polygon, similar to what is used when solving equations using the simplex method. See figure 4.1.

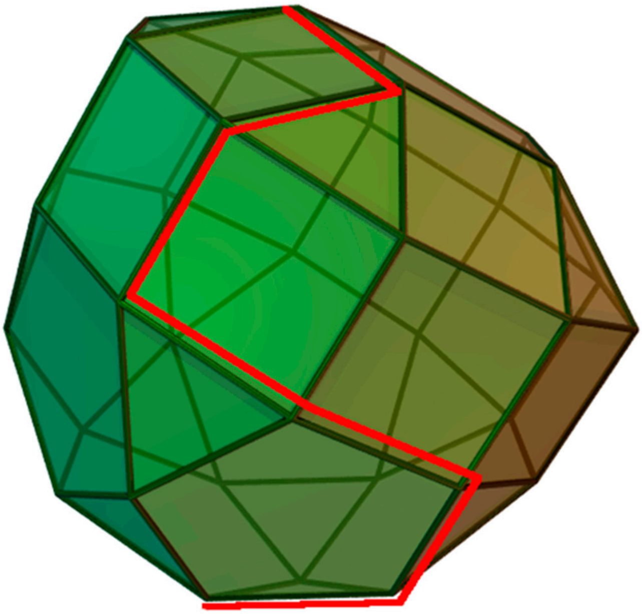

Figure 4.1 Multidimensional polygon. Figure from Wikipedia.

The function state matrix is a practical way to describe how automation systems limitations are designed in real life. It does not have to be complete, since most systems only handle the relations between inputs and outputs that are subject to control (at least the ones that have commercial interest). Also, it does not differentiate between continuous and discontinuous processes, and functions with discrete values can be freely added. However, a discrete manufacturing system will only allow solutions to the equations that are integers (diophantic equations). The same applies to event-driven manufacturing processes. The function state matrix is a general solution to the problem of describing how outputs are influenced by inputs and it does so in a way that lends itself to computer programming.

To illustrate a key aspect: if a pharmaceutical process produces 12 liters of vaccine the process designer may be interested in control that can produce 12.5 liters. However, in an assembly operation that produces excavators the designer is only interested in software that makes 12 or 13 units per day. Solutions that solve the problem of making 12.5 units are discarded.

If a parameter reaches a boundary condition, the process will stick to the boundary but the process can move along the boundary surface if the parameter changes. This model does not take into account cases when the boundary areas are nonlinear, “bulging”. However, it has been judged that this limitation is of little practical importance. If necessary, some “thickness” on the boundaries can be accommodated by using a dead band around the variables.

4.1 Measurements

Correct measurement of input and process output signals is the foundation for all control actions. Traditionally the signals have been divided into digital (on/off) and analog variables.

Digital signals are easy to understand in principle, but also contain ambiguity. For example, if one uses 5VDC logic, then a zero voltage would be interpreted as OFF and a 5V signal as ON. However, the electrical circuitry may decide that all signals between 0 and 1.8 V are OFF and all signals between 3 and 5V as ON. The consequence is that signals between 1.8 and 3 V are ambiguous.

A serial communications line is a simple example. The signal strength decreases with the length of the cable. If someone adds a bit of cable length or the contacts corrode, signal ambiguity can be introduced.

The same situation appears for analog signals. When an analog signal is converted to a digital value it is often constructed via comparison with a voltage slope. The result is a digital value with a certain accuracy, such as a 16-bit value. There will be a rounding error depending of the resolution of the A/D converter.

The rounding error can in most cases be discarded, but not if it is used in a recursive algorithm.

Recently, more signal categories have been used for control purposes. These include images, patterns and composite values like vectors. This field is developing fast.

One example of the use of images in control is inspection. After the paint line in a car or truck assembly, a digital image of a bonnet or a door is made. If the bit pattern shows scratches or impurities, the part is sent back for rework. The reason is that the human eye is very apt at spotting paint damage, therefore chances are that a vehicle that does not have perfect paint coating will be harder to sell.

Another area of interest is model-based measurements. If a system model is available, a measurement value can sometimes be calculated from the values of other sensors. An example of this is flow rate, which often is hard to measure with high accuracy, especially when the media velocity is low. Instead, a pressure difference can be measured if a constraint is introduced which alters the velocity of the fluid. One reason can also be that the sensor for the desired quantity is expensive, and other sensors are more economical. One example is energy flow in small heating distribution systems, such as those for apartments. The cost of the instrumentation may not justify the potential savings in energy by implementing control.

There are also measurements that are depending on each other, for example readings that are dependent on movement, detected by an accelerometer. This is often the case when a smartphone is used as a measuring device.

Example 4.1 *Successive measurements*

> There are cases when successive measurements are taken to identify a process parameter curve, where the last part of the curve cannot be measured directly for some reason, for example a too high environmental temperature, e.g. in a steel furnace. Because the temperature measurement is incremental the successive values contain some of the old rounding error and if the process is iterated too many times, the identification will be inaccurate.

Complexity can be created by several actions originating from the measurement of signals.

Sometimes the signal value is complemented by a VTQ parameter (Value, Time, Quality). This enables the user of the signal value to take decisions (like discarding) on how to use the value. A simple way is to have max and min values corresponding to the physical limitations of the process. An early way to implement this type of checks was the change of 0-20 mA to 4-20 mA where in the latter case 0 mA would correspond to a faulty sensor. The accuracy of 4-20 mA is adequate for the vast majority of process sensing needs. The original published standard for 4-20 mA, ISA SP50, was first published in 1966, over 50 years ago, generating widespread acceptance and creating an installed base of millions of points. Electricians find it easy to wire and troubleshoot 4-20 mA. A description of the SP50 standard can be found in the Instrument Engineer´s Handbook, edited by Liptak (2002).

Example 4.2 *Event spawning*

> Event spawning occurs when a control signal is duplicated and used for an action based on a threshold value. Then, two actions can be taken/caused based on a single event.
>
> If an analog value is copied to two A/D converters and compared in two different control systems, then a small difference in conversion accuracy will cause the control systems to create two actions based on a single event (a passing of a threshold). There can also be a time delay between the two actions.
>
> This situation is becoming more and more common, since the installation of bus systems on the factory floor means that there could be many users of the same data value.

Another development will accelerate this development since many sensors are nowadays equipped with processors, so that the measurement signal is converted at the sensor. This eliminates analog errors in signal transmission, but creates new sources of complexity. If a sensor is being calibrated or otherwise subject to some action that causes the value not to be representative for the true signal reading, then this situation can no longer be handled at the control system level. If the situation is not properly handled, it will result in the signal being outside the control boundary.

New developments can also create confusion in existing system structures. One example is sensors that can measure different types of parameters depending on commands sent to them over a communication line.

4.2 Effects of the law of big numbers

Many sensors nowadays have processing power. It has been increasingly common to make a series of successive measurements and improve the accuracy by using exponential filtering, averaging methods or other filter algorithms to eliminate sensor noise. The law of big numbers says that by averaging 100 measurements, you gain one decimal point in accuracy, if the errors are distributed according to a Gaussian curve. A description of the normal (Gaussian) distribution can be found in the Encyclopedia of Mathematics by Hazlewinkel (2001). The principle was first discovered by the mathematician Giordano Cardano, but the name of the law was given by Poisson (1837).

However, a correct interpretation of a value constructed this way is that it is ***static*** during the time it takes to collect the values. If collected during acceleration or deceleration, it will give a false derivative for e.g. velocity. The same is true for flow in pipes.

Also, with the proliferation of wireless data collection, it has been increasingly common to activate/deactivate measurement sensors prior to a transmission. This is done in order to conserve battery capacity. The sensor is powered up and the processor will make a series of measurements and send when the value from the sensor gives a stable output signal.

This is also a potential source of complexity, since it almost assures that the changes in measurement values from one time to another will never be continuous.

4.3 Control signals

Controller outputs are categorized in the same way as inputs, as either digital or analog. There are also variants of digital, such as pulse outputs and analog values realized as pulse trains and other formats. Control signals are influencing the controlled process through actuators. The signals suffer from the same types of ambiguity as the inputs. However, with analog control signals D/A rounding errors are seldom a problem. A much more common problem is actuator nonlinearity, i.e. that the control range contains setpoint subranges where the actuators react with instability.

Another common problem is when control signals are failing to achieve the desired setpoint. In PID control, the integration part of the algorithm remembers past noncompliance and if the system has been unable to reach the desired setpoint for a prolonged time, then the system will take a correspondingly long time to return to the regular controlled signal span. The compensation for this effect is known as anti-windup in control loop controllers.

The most common problem with signals that cause complexity is, however, that data have a limited validity period. A measurement data value is valid until the sensor has provided an updated value. Since control signals always operate with a delay from the input signals, they are often based on old input data. If the process dynamics are fast, then this delay can be serious and sometimes one does not trust the system to react quickly enough so the input signal to actuator loop is closed almost at the sensor level. This is the case in many vehicle control systems.

When a value on a control bus is complemented with a VTQ (Value, Time, Quality) parameter it enables the control system to determine if the value provided can be trusted. However, this also generates new sources of complexity: what should the control system do if there is a problem with the value (use the old one?, for how long?). There could also be complexity in interpretation ambiguity since a problem can have multiple causes and the true origin could be obscured.

When sums of values are accumulated and compared from different time periods, illogical results can occur, such as a process operating over 100% of capacity. One such example is described in Example 4.4. Time lags in process ERP (Enterprise Resource Planning) systems can frequently create measurement errors of this type.

Some systems cannot be taken off line, for example dam controllers. If it rains and the dam fill up to capacity but the rain does not stop, the system will and must fall back to a ***degraded*** state of control. It is for these cases that process modelling can be of great help. If a model predicts degradation of operation with current settings in a certain time, then actions, like the changing of setpoints, can be made in order to save time. Unfortunately, one type of degraded state is a complete system failure.

4.4 Actuators

Actuators are the physical units that convert the outputs into physical actions (valves, motors, heaters …). There are two main areas of concern with regard to complexity. The first one is the fact that while there are many types of measurement units, the number of actuators is much smaller. Therefore, we cannot in many cases influence a desired variable directly, but we have to influence it through variables for which there are actuators. This is often caused by economic constraints. For example, controlling small flow rates may be costly, so maybe the process has been designed to use the level in an intermediate vessel instead.

The other big factor which influences the level of complexity is the fact that many actuators are highly nonlinear. A valve for a domestic radiator may provide 50% of output variation in the first tenth of a full turn. Similarly, flow control of compressible media, such as air, is nonlinear. Again, this is also influenced by economic constraints. We have to act through the control model but there may not be enough funds available in order to afford sensors in sufficient numbers to calculate all the interesting variables from the nonlinear equations.

Example 4.3 *Air supply in wastewater treatment*

> A wastewater plant had three aerator compressors, where two were fixed speed and one variable speed compressor. The idea was to use the fixed speed compressors as a base supply and the variable speed compressor as the swing supplier. Control was based on the DO (dissolved oxygen) concentration in the activated sludge reactor. However, at certain setpoints the result was that one of the fixed compressors were started and stopped repeatedly because the combination of desired DO setpoint and actual flow could not be controlled

with the actuator configuration at the site. Similar problems will also appear in pumping systems using fixed speed pumps.

Early implementation work in this area was done by the author and G. Olsson (Gillblad and Olsson, 1977, 1978).

4.5 Homologation of data

Automatic control is an information (data) driven process. In the first implementations, information transfer was mechanical and a breakdown of information corresponded to a mechanical failure. In the successor, the analog machine, information transfer was also almost immediate but these days, with ubiquitous digital processors, delays in data processing are inevitable.

This is an area of growing concern and it greatly adds to complexity. The effect is easiest seen in manufacturing, which normally contains much faster processes than the process industry. The actuators that exert digital control often transition when the alternating voltage passes zero. In a 50 Hz system this means 100 times per second. As a consequence, most PLC systems are designed to have a cycle time faster than 10 milliseconds so that most digital inputs state changes will be recorded within one PLC duty cycle.

If data from different time periods are compared and control decisions taken based on data that in reality have changed due to too slow updating, errors will occur. Unfortunately, this problem is often ignored and is a serious limit to building larger integrated systems as well as a common cause for projects not to meet the desired performance goals.

Control systems are at the base of the data structure pyramid (see Chapter 8.2) and the idea is that the systems at the bottom are characterized by large number of data transactions but only few data elements in each transaction. As data flow upwards in the pyramid, it gets successively more aggregated and at the top one finds files and large database structures.

In order for systems to draw correct conclusions from the data, it is important that data from the same time periods are compared, otherwise errors will occur. The process of treating data so that they become comparable to each other is called homologation.

Example 4.4 *Battery manufacturing*

> Manufacturing of lead batteries is a time consuming process. In principle lead batteries are similar in shape and capacity, but there are still many types of models. First the plates are formed, then they are placed in a container and the poles soldered. Later, acid is added and finally the battery is charged. Because the process takes several hours, the traditional method was to empty the production line before a new type was introduced. A company rationalized the process in such a way that only one machine was stopped at a time and the new battery type was introduced while the previously manufactured type was still in the line. The production volume increased because fewer machines had to stand idle during the product change but the ERP system (Enterprise Resource Planning) at the top was not designed to handle two products at the same time in the same line. When data was homologated, the net result was that production exceeded theoretical capacity (based on a minimum product change time).

The manufacturing of batteries is currently a fast growing industry. The energy density of batteries has increased significantly. However, high energy density means that a short circuited battery risks self ignition by overcurrent and thus become a fire hazard. In order to handle this situation, modern batteries are designed as packs of smaller units and control is introduced, so that the extraction of energy is slower than at charging. This is another example of integrative complexity (see Chapter 6.3).

Photovoltaic systems often combine low voltages with high currents. Since switching high DC currents is electrically nontrivial, batteries are often wired in series in order to reduce the current. The design trade-offs in this area are currently subject to intense research and the aging of electrochemical compounds by successive charging and recharging is a good example of operational complexity.

5. Industry examples

This chapter summarizes a number of industrial examples that are analyzed to characterize and illustrate the complexity issues. This leads to a suggestion of a formal theoretical approach to the different production areas in Chapter 6.

Example 5.1 *Wastewater treatment plants*

There are numerous types of wastewater treatment plants. The general characteristics are that a wastewater stream enters a biological section, where bacteria consume the biodegradable material in the influent stream. The bacteria are then recovered through a sedimentation process followed by a recycle and the remaining contaminants, such as phosphorous components, can be treated with chemicals, like ferrous or aluminum based additives. Energy can be recovered through biogas production and sludge stabilized via dewatering and polymer additives. A more complete description of the processes is found in a textbook by Henze et al. (2002, 2008). The complexity in a wastewater plant comes from several sources. Firstly, the raw material (sewage) is not precisely known. The quality and quantity varies over time. There is a diurnal variation due to people´s daily habits, rainfall will dilute the concentration of biodegradable material, spills from industries, backwashing from effluent treatment and from drinking water plants, as well as singular events (like the first rainfall after a dry spell will create a surge in influent suspended solids).

It is not economically feasible to build wastewater treatment plants big enough to handle all these variations, and it is common practice to design for the maximum flow that a plant can handle as three times normal average flow. If it is bigger, excessive flow can be bypassed or partially treated in order to save the microorganisms in the system from being washed out. The uncertainty of input creates complexity of operation.

Another area of complexity is that the biological process itself is not completely known and the microorganism composition may change with the quality of the influent wastewater. Mathematically, the control system is to be described as a set of equations with too many unknowns and too few measured variables required to solve the equations. In practice, this can lead to growth of bacteria of unwanted properties. One example of this is when the concentration of biodegradable material is low, then filamentous bacteria with a higher surface-to-volume ratio are favoured at the expense of more round ones. Unfortunately, such bacteria do not settle well and the sedimentation process is not designed to cope with floating bacteria. Other such processes exist, like flocculation and decanting, but no single process can handle all possible process states.

It can be safely assumed that the public would like to see the cleaning of used water to be as efficient and as predictable as possible. Adding new process steps only increases the process complexity. In order to remove ammonium-nitrogen the aeration basin has to be enlarged and anoxic unit processes have to be added. Anoxic processes are processes that do not use oxygen. To remove phosphorous biologically then another anaerobic process has to be added. However, there is a stoichiometric relation between base materials. For example, carbon is required in order to reduce nitrate to nitrogen gas in the anoxic process. The mass balance requirements may not be met due to temporary or permanent shortages of certain ingredients.

As a result of this, one could conclude that a wastewater treatment plant cannot be adequately controlled. In practice, this is not so. The reason is that in biological systems, if one process fails another is likely to take its place. However, if some basic conditions are not met, like the availability of dissolved oxygen, then the process will be unable to provide any desired output.

On top of this, there is a sequential relationship between the processes in a wastewater treatment plant. If the flow rate is low, then the pre-sedimentation will remove more biodegradable material from the aeration section. As a consequence, some of the air supply energy may be wasted. If the influent flow rate is high, then the chemical precipitation at the end of the plant may become inefficient as more chemicals than necessary will be dosed compared with the remaining suspended materials.

In traditional design practice, the complexity issues in a wastewater treatment plant have not been met by detailed dynamic analysis and careful design of control systems, but by oversizing. The theory behind this is that if the plant is big enough, then all the variations through external forces of biological process variations could be "dampened", simply by making volumes larger. However, during the last decade an increasing number of treatment plants are designed using detailed dynamical models of the hydraulics, biological kinetics and sedimentation.

Recently, new requirements have been added to what a treatment plant should be able to do. Some stable chemicals like phtalates and pharmaceuticals that are ingested have been accumulating as a result of recursive complexity (see Chapter 6.4). The risks involve threat to reproduction and of DNA corruption. Unfortunately, complexity has a tendency to reappear somewhere else if attempts are made to reduce it in one area.

The wastewater treatment plant is mainly a functionally complex operation, but it also has operational complexity aspects.

Example 5.2 *Water savings schemes*

> In some regions, water tariffs have been designed in order to encourage conservation. However, the wastewater pipe grid has been optimized with a certain sloping angle, designed for a trade-off between flow rate and the number of lifting pumping stations required. When people reduce their water usage, then flow rates are reduced and sedimentation will start in the sewers, which results in odour, clogging and uneven pumping. Because of this, sometimes additional injection (extra water) has been necessary in order to restore flow.
>
> The same problem can appear if the water administrators try to use the storage capacity in the pipe sewer grid to increase the maximum hydraulic sewage load that a wastewater treatment plant can handle. The design load is called Q^{dim}. By adding sewage sluices and controlling pumping the input flow can be dampened during rainstorms thus reducing the risk of bypassing. However, this procedure can also cause sewage sedimentation in the sewers as well as increased risk for generation of odour and emission of greenhouse gases (gases that increase atmosphere temperature) like methane and nitrous oxides.
>
> A similar problem can occur in the distribution of water for consumption. Water savings will mean that the capacity in the distribution pipes is larger compared with the average consumption flow, thus increasing the "age" of the drinking water, affecting its quality. More cleaning agents, like chlorine, may have to be added. Fungi are not so affected by chlorine, which may require new treatment steps to be added, like treatment with ultraviolet (UV) light, in order to reduce bacteria.

The water distribution is mainly an operationally complex system.

Example 5.3 *Petrochemical crackers*

A petrochemical cracker is the main process in a refinery. It is surrounded by off-site auxiliary processes, such as steam production, hydrogen production for hydrogenation, tank farms, oil movement, blending and flaring, as illustrated in Figure 5.1. Oil movement is centered around tank farms. Some tank farms can be very big, containing over a million tons of petroleum products. Since this represents a large commercial value, automating the oil movements is a priority. A suggestion on how to do this automatically was presented in 1985 (Gillblad, 1985). The development was part of work done by the author as member of the International Organization for Standardization (ISO) committee for hydrostatic measurements in petrochemical tank farms.

Subsequently, a series of application dedicated (flameproof) PLC´s (Programmable Control Systems) were developed, and later installed at the Esso Fawley refinery outside Southampton. The system was comprised out of some 212 PLC´s of which 202 were in the hazardous zone, controlled via 50 kilometers of fiber optic communications and a pair of redundant (double systems in order to increase reliability (uptime)) computers. The author served as project manager and reported the results of the installation at a conference in Oslo, Norway (Gillblad, 1987).

The main process in a refinery is the cracker. In short, the cracker is fed by raw material, feedstock, which then is fractioned according to its properties, with lighter fractions at the top and heavier at the bottom.

Cracking can be performed in steps and various streams can be siphoned off to be further processed into more definite hydrocarbons, according to octane or cetane numbers.

The complexity of a cracker arises from mainly two sources, both connected to the main purpose of the cracker, which is the ***yield***.

The yield is the desired output mix, which is heavily influenced by the contents of the feedstock. For each feed stock and each process there is a range of possible control scenarios.

Besides being functionally complex, the cracker has an operational factor. Since the density of most hydrocarbons are heavily dependent on temperature, the difference between seasons can affect the control. This is easily demonstrated for a car. If a petrol tank is filled up completely in a vehicle and is let to stay in the sun for a couple of hours, chances are that the tank pipe inlet will overflow.

If the control of the petrochemical process was based on weight instead of volume, then this problem would diminish. A design proposal was presented at Het

Instrument by the author in Utrecht, Holland (Gillblad, 1988). Practical trials done at the Elf Donges refinery in France were reported in Nice, France (Gillblad, 1989).

Current product development is centered around making petrol with tighter specifications in collaboration with motor designers. This creates a more effective combustion, which results in higher output per cubic centimetre of engine volume and lower pollution.

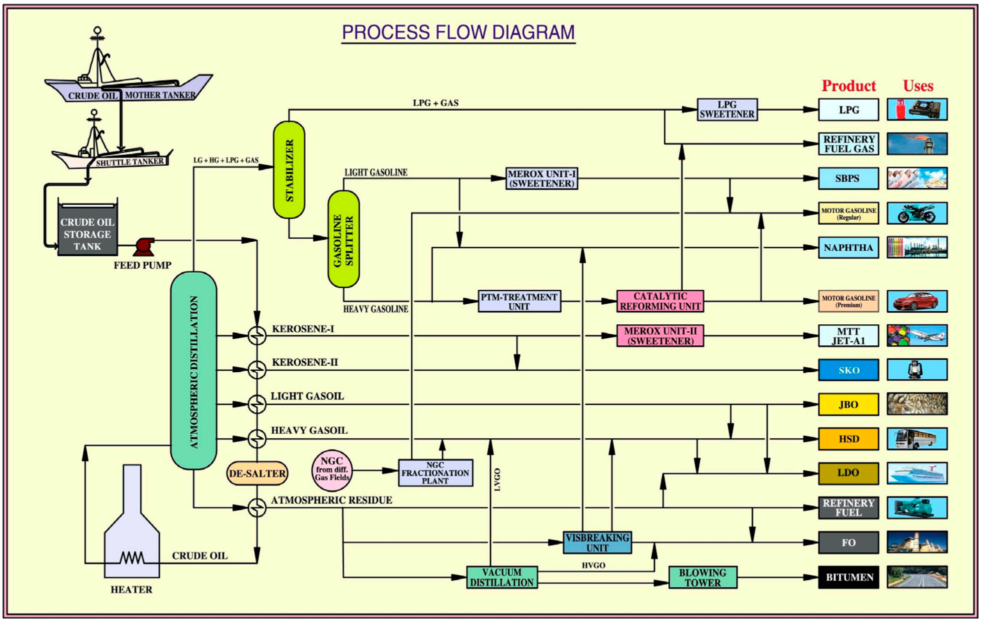

Figure 5.1 Refinery process layout. Picture from Ireland News.

Example 5.4 *Production of lubricants*

> At a refinery specialized in lubricants, the management made a business decision to buy less expensive feed stock on the spot market. This was not of the same quality as the standard feed stock, which resulted in lower quantities of the desired high-quality lubricants. Unfortunately they failed to inform the sales department, who had already sold the projected quantities. To increase production to meet the volume sold would have meant even higher production of unwanted qualities, so the missing volume had to be bought at unfavourable market prices, which resulted in an overall significant economic loss.

The main complexity source is that the process itself is large, and cannot be easily divided into smaller subsystems. Many chemical reactions are reversible, and the total number of possible reactions is significant, although the preferred ones are favoured by using catalysts. The variability of the input can also be limited via analysis of samples.

The first automatically controlled cracker contained about a dozen PID loops but a modern unit contains many hundred. Feedstocks can be modified by hydrogenation and the process yield optimized by specific catalysts (zeolites). The result is that operational complexity is greatly increased.

The goal of most cracking operations is, besides optimal yield (as much as possible of the most valuable compounds), a reduction in variability. Products destined for combustion, such as airplane kerosene and high octane petrol, command a premium price if designed for optimum combustion without pollution and engine torque maximization. Control towards more exact setpoint control increases complexity.

Example 5.5 *Electronic components production*

Electronics components production is similar to production of pharmaceuticals in the sense that the raw materials is not a big cost, but the manufacturing process and the associated engineering dominates the investment.

In both these industrial branches, some manufacturing processes have the highest value added factor in a single process step in the whole industrial sector.

Most electronics manufacturing starts with a slice of very pure silicon, cut from a large single grown crystal. The shape is circular, and over time the size has grown, from 100 mm diameter to 200 mm and further to 300 mm. At the same time the possible individual integrated circuit sizes have shrunk, thus enabling more products to be made in each process cycle.

The circuit pattern is etched onto the wafer by photolithography, and then the doping agent is diffused onto the exposed parts of the silicon. After optical inspection the integrated circuits are packaged as chips. Sometimes several chips are put together on a chip carrier and there is also a recent development creating microscopic three dimensional circuit assemblies using LDI (Laser Direct Imaging). The company Mycronic has a machine doing this and the process is described by Barbucha et al. (2008).

The author was contracted in 2015 to help with implementing communication to a new machine, which uses a high power laser to create the ability to build processors with many cores. In two dimensions, the area becomes congested at about 8 cores, but there is an interest to create processors with many more. The new machine has significant functional complexity as the laser beam is controlled by a multicontroller computer with more than 8 000 analog outputs.

Like in petroleum cracking based on catalyst, the yield is the main process parameter in an electronics production plant.

New types of more highly integrated circuits command a higher price before they become more commoditized, and the number of units per production run is higher. Initially the yield is normally lower, and the process to increase yield in a given time frame is usually hectic. As soon as the optimum yield is reached the debugging process is stopped. If more products are needed, an ***identical*** plant is built in order to reduce the time to achieve optimum yield in the second entity.

The complexity issues in an electronics production facility are mainly related to conformity. The production process window is narrow, meaning that the setpoint variations must be very small. This manufacturing process is mainly functionally complex, as the boundary conditions are narrow.

Example 5.6 *Water spraying in an offset press*

A printing press is a big machine, with either an exactness requirement (fine grade print) or a production requirement. The discussion here is limited to the offset press, illustrated in Figure 5.2.

In an offset press, the image is introduced on the rolls via a cliché. There are normally three colour rolls and one black. The paint has a high density (thixotropic), because the production speed is largely dependent on the drying time. A thixotropic fluid is a fluid which takes a finite time to attain equilibrium viscosity when introduced to a step change in conditions. Many gels and colloids are thixotropic materials, exhibiting a stable form at rest but becoming fluid when agitated. If a press is run too fast, then the images will be smudged. Previously this was addressed by using paint solvents instead of water, but because of health concerns this practice has been stopped, and most types of printing paint are water based nowadays.

After the print process, the printed material is treated to fit its purpose, cut into the right size, stapled, put on pallets for distribution etc.

The complexity in controlling a printing press comes from the process itself, which is very fast. A big printing press that makes commercial leaflets can print one million copies during one shift of operation. Most setpoints are proportional to the rotational speed of the rollers, and the controllers have to be updated very fast, like in the example below.

It is possible to increase production in an offset press by increasing the velocity of paint absorption into the paper. This is done by very precise spraying of water just before the paint is applied. When a new print run is started, the print rollers accelerate and more water has to be applied in order to maintain correct surface moisture level. Every spraying unit needs new setpoints for this, which is not so difficult when the press accelerates but more critical when the run is to finish (the rollers decelerate). The reason is that if too much water is applied, the paper will become too moist and break. How many setpoints are needed? Typically, a big press (a million copies per 8 hour shift) will need about 10 000 new setpoints per second.

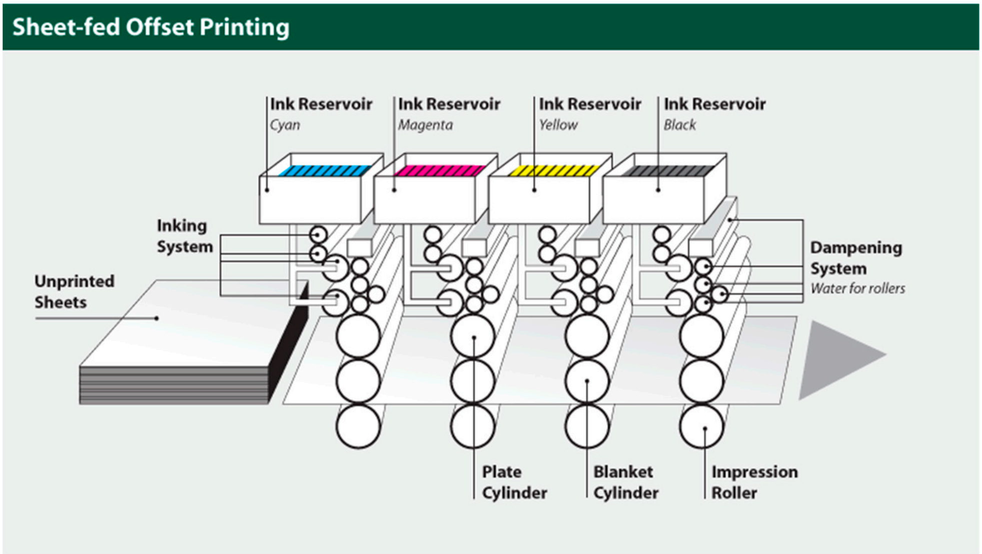

Figure 5.2 Offset press layout. Picture from Pinterest.

The complexity here is related to the fact that this is a highly integrated mechanical process, so the process cannot be divided easily into smaller units. Each controller is coupled mechanically to the neighboring controllers and the correct operation requires meticulous synchronization.

One side effect of a high level of automation is the possibility to create costly mistakes. On April 2, 2018, the Swedish newspaper Svenska Dagbladet erroneously printed and distributed a full wrong edition of the paper (from a day earlier).

Example 5.7 *Ore treatment plants*

An ore treatment plant is a big physical process. Different ores are treated differently so the discussion here will limit itself to metal ores, in particular iron ore, which is the largest metal raw material in volume by an order of magnitude.

Traditionally, iron ore was transported to the steel works where it was crushed and treated. Since iron ore rarely has more than 60% iron content, a lot of stone material was unnecessarily transported and the prevailing method now is to pretreat the ore at the mine and send pure steel directly to the steel mill where it is converted to steel by direct reduction. At LKAB (Malmberget, Sweden), various steel ore qualities are produced as pellets, making the direct reduction steel production process even more precise as different pellet types can be used for the production of various steel qualities.

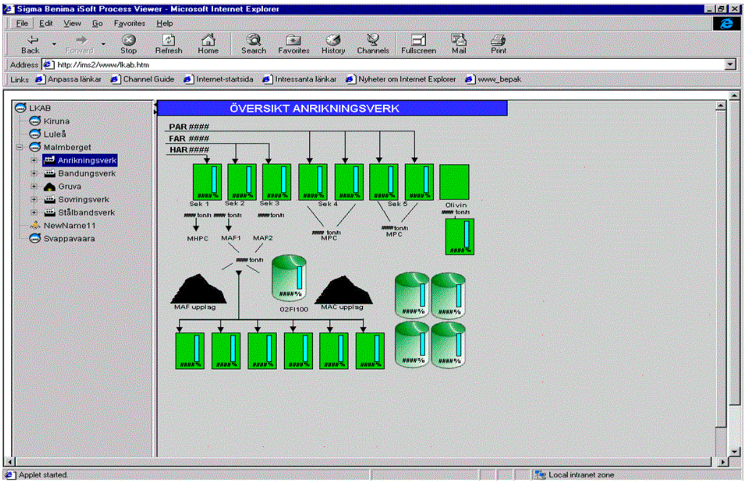

Figure 5.3 Screen shot from control overview of the steel pellet pre-production at LKAB Malmberget.

In principle, this process is a steel pre-production process in order to produce intermediary products. A screen view of the information presentation system of this process is shown in Figure 5.3. In theory it is not complex to understand but because it is governed by some parameters that cannot be reasonably controlled and only to a certain extent measured, the process contains inherent complexity. The reason is that the quantities are large and the stone/iron relation in the ore varies from where in the mine the ore has been taken.

The mining companies take samples in the mines of this relation, which is forwarded to the processing units, located at the surface. The transport time of the ore causes a long lag in the supervisory control algorithm, as there is no identity tagging of the sample vs. the actual quantity transported. One has to assume that the sample is representative for the ore in that part of the mine. The lag problem is further affected by the fact that there could be extra delays in the transport chain due to breakdowns, resource shortages or other reasons. The consequences are magnified by the size of most iron ore production sites.

The ore production complexity is mainly of the operational type. Extracting ore is functionally simple, as it is one of humanity´s oldest production operations.

Sample calculation. An annual volume of 20 million tons represents 650 kilos per second. A time lag of two hours means that a variation of one percent in the iron/stone relation represents 47 tons of iron, which is a full truck load. This is important to take into consideration in the planning of how many trucks or drivers that will be needed.

Example 5.8 *The steel rolling mill*

> A rolling mill is a plant which produces steel in rolls from raw material from a steel plant, called ***billets***. There are hot rolling mills and cold rolling mills, depending on if rolling is made under or above the crystallization temperature of steel. The hot rolling mill is more common and will be described here.
>
> A hot rolling mill is composed of a series of processes, starting with a furnace, where the billets are heated, Figure 5.4. The billets can also come directly from a continuous caster, but we will consider a case where the billets are heated.
>
> The hot billets will pass a series of rollers, which reduce the thickness of the steel at progressively higher roll speeds. When the desired thickness has been reached, then the steel is placed on a roll in an unwind stand. The steel surface oxidizes at high temperatures, called ***scaling***. The scales are removed in a descaling process, and in most cases reused by being remelted.

The complexity in this process manifests itself in a conflict of interest between running each process step according to its own characteristics. For example, the oven is normally run according to energy conservation criteria. The rollers must be run against a tradeoff between thickness reduction and roller wear, which is measured as dynamic pressures and motor speeds. The descaling should be done with a minimum loss of material.

A normal oven contains several billets, and if the billet is to be made into a thin product, the starting temperature has to be higher, since a higher milling speed means faster cooling of the product. However, a higher temperature in the oven means higher energy consumption which would be wasted for thicker products.

This is an example of Cyber Physical Systems of Systems, further described in Chapter 6.2.

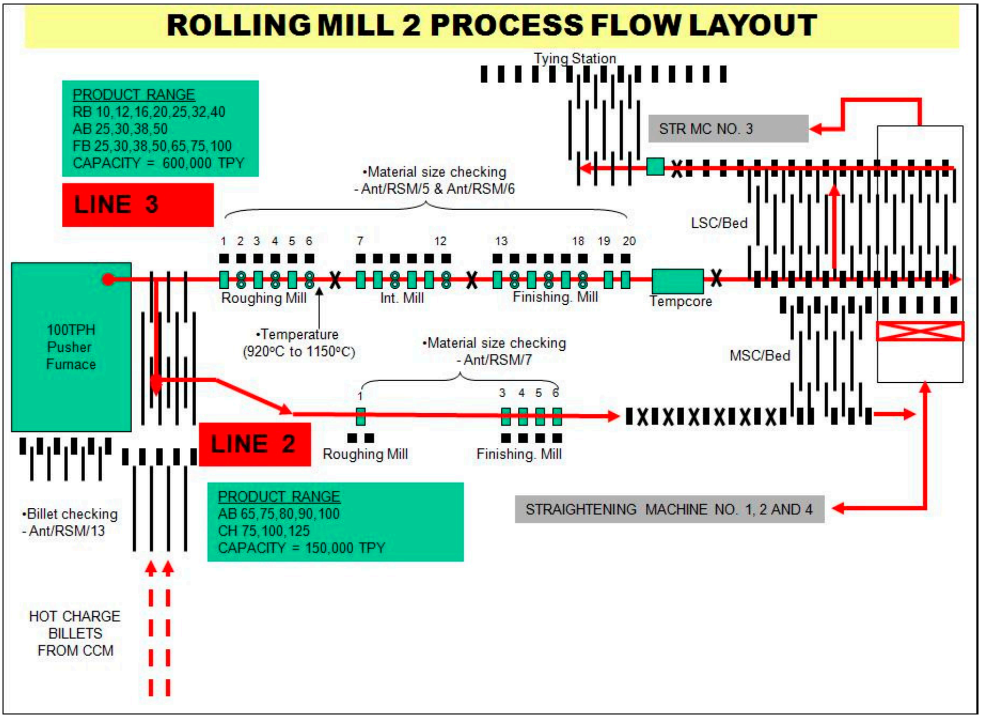

Figure 5.4 Process flow overview of a rolling mill. From Primetals Corp.

Sample calculation. The overall yield in a rolling mill is measured as the volume of raw steel necessary in order to make a ton of finished product. For example, a mill run of a 30 ton billet could mean that 29 tons of finished product would be produced. The finished product needs to have its edges cut and sometimes be grinded to achieve flatness. To further complicate matters, the yield loss cost could be different if the mill was running at full capacity or if it had capacity to spare.

Example 5.9 *Impact of plant operating experience*

The steel industry is an eminent case where experience in running a plant guides its future operation. Even if the control models of the various processes are not perfect, most rolling mills have excellent statistics on yield. This offers a rare opportunity to assess the cost of complexity.

Let us assume that a rolling mill has records from 1000 runs, where the yield (in successful runs, not runs where scrap metal has been produced) has an average use of 1035 kilos of raw material per ton finished product with a standard deviation of 10 kg on the billet weight. The product usage range will then be anywhere from 1025 to 1045 kilograms. The conclusion is that the mill has a technical capability of running at a minimum loss of 2.5 %, whereas 3.5% is the average resulting loss. The complexity in this example costs 1%, which theoretically in a mill with a yearly production tonnage of 3 million tons would represent an extra reworking volume of 30 000 tons.

These types of calculations can be used to calculate the cost of complexity.

Example 5.10 *Automated wooden house factory*

The building industry is a sector that has resisted automation to a large extent for various reasons. One exception is the production of single family wooden houses, where the automated production volume dominates the "hand-built" houses by an order of magnitude. The reason is cost effectiveness.

A wooden house factory is a semi-automatic process, where house parts ("elements") are produced by workers in successive assembly stands. A modern factory can contain up to 30 such stations. The control is open loop, where the instructions for each station are received based on the order of the elements passing the factory. The original "production recipe" is a house design made by 3D CAD software, which is then converted into x,y,z positions. These are then used by the automated part of the production equipment, mainly nail guns.

The elements produced are of various types, such as floors, roofs, outside walls and inside walls. The last station will assemble these into modules, which are of a size commensurate with the size allowed for road transport.

Compared with many other automated manufacturing processes, the automated house factory can be considered "simple". However, it is the complexity inherent in house building that impedes further rationalization of this industry.

The reason is that the housing plant is constrained by its limited range of control authority. Simply put, it is difficult to meet all the regulations that govern a housing product and still have a suitable product for automation.

Example 5.11 *Manufacturing constraints in-house production*

Most housing plants are constructed to make walls with a maximum thickness (e.g. 453 millimeters). The maximum width allowed for transportation on Scandinavian roads is around 3 meters. The tradeoff here is: the thicker the wall, the smaller the living space of that module will be. Government may specify a maximum use of active power allowed for new houses, for example 9 kW for a one-family house. This may in certain cold regions require thicker walls than can be automatically produced.

Another problem is the wood itself, which is a "live" material. Older beams could bend or warp, especially in lesser dimensions and if they have been kept in storage with high humidity. Also, older trees have a different density and even the position in the tree where the plank has been cut has an impact since trees really are shaped as cones with a growth direction (except when using hardwood like mahogany). These differences can cause the nail guns to miss their target points, which reduces structural strength.

As a summary, the complexity in an automated housing manufacturing facility comes from the fact that it is not easy to make a production recipe that is conformant with all constraints pertinent to house construction. If a window is designed too close to a corner, the tensile strength of available beams that are available at approved dimensions may not exist. Metal beams could do the job, but the factory may not be able to handle metal fixtures. Complexity reduces significantly the number of products that this type of manufacturing can handle.

Recently, the technology of designing houses has reached a level of sophistication that enables the designer to construct a house that can be assembled completely automatically in a factory (everything except the foundation, but all installed apparatus like kitchenware and heating systems are included). The building constraints are checked by the CAD software so that the end result is a "recipe", which is a complete house structure that can be downloaded to a suitably automated factory. Assembled parts that require drying time, like bathrooms, are made as subassemblies and placed on pallets along the assembly line.

Example 5.12 *The electric power grid*

It is well known what an electrical power grid provides. It is one of the most important infrastructures in a modern society. Therefore, the correct functioning of the power grid is essential. At one end we have the power plants, which supply electrical energy to the grid and at the other side we have the energy consumers, which use energy at different voltage and power levels. The supply contracts vary from customer to customer, but a common industry contract would involve a number of kilowatt hours at a certain ampere level. The consumption varies over time, and the energy providers want to limit the reserve capacity at the grid´s disposal, since this is a cost without corresponding income. Therefore, large contracts often involve a penalty clause, where consumption over contractual limits incur stiff penalties. There could also be clauses where a large user limits consumption of energy in times of relative shortage. This is called *load shedding,*

Furthermore, some delivery contracts are based on maximum average prices, so that a temporary shortage can mean that a single hour of poor delivery capacity can affect costs over a long period. This is done in order to reward customers that implement load shedding procedures.

> At the Sibes foundry in Gnosjö there are three brass induction ovens. The company had a contract with the local power provider that specified the delivery of a certain number of kilowatt hours at 500 amperes at a fixed price, but a much higher price if more was used. A system was installed to calculate the amount of energy used per hour and if the number of contracted kilowatt hours was expected to be exceeded that hour, the control logic would first cut the power to the oven that had the highest differential between actual temperature and the minimum setpoint temperature. This shows that boundary condition constraints may be economical as well as physical.

Providing electrical energy is an industry where "big is beautiful", at least from a production cost perspective. A big grid means that each consumer connected to it is relatively less important, and the spare capacity can be smaller as a percentage of the total capacity. Therefore, grids have become more and more interconnected over time, spanning over thousands of square kilometers with thousands of transformer stations. However, since each transformer station has a finite reliability, the probability that the whole grid is completely functional at any one time decreases. Also, some grids are designed to operate in a degraded mode, e.g. all parts does not have to be operational at the same time.

The complexity in an electrical power grid arises from the size of it. In a small system the number of rare occurrences can be ignored, whereas in a large system, it cannot. In a small system each unit event can be proportionally more important, and thus the risk that the whole system is affected is greater. There is a tradeoff between the optimal operation under normal circumstances vs. the potential catastrophic consequence of a rare number of events. This is a complex design criterion, and the execution of a strategy in the event of a rare occurrence is even more difficult since the sequence of events in a power grid is so fast.

A compilation of the current state-of-the-art in power control is provided in the PhD thesis by Svensson (2006). The risk of the market, e.g. energy pricing, having an impact of the physical control of power grids have been studied by Li et al. (2017).

The power grid is an example of integrative complexity, although there are some operational complexity areas, mainly when dealing with various types of exception handling when the grid is outside its normal area of operation.

The functional complexity aspect centers around the fact that the consequences of an error like a substation trip can propagate almost as fast through the grid as the information about the trip.

Example 5.13 *Heat pumps in private homes*

Many owners of private homes are installing heat pumps in order to provide tap water and heating for their household. In essence, this process converts about 1 kWh of electrical energy to 4 kWh of heat, somewhat depending on which media the energy is extracted from. However, the electrical energy used for heating is often a lot larger than the municipal grid was designed for originally, which is domestic consumption such as ovens, lighting, refrigerators and TV sets. The energy demand varies over time, with a maximum during winter months in cold countries like Scandinavia. Each subscriber normally has a clause in the supply contract, which allows the provider a variation in the quality of service, like 240 V plus 10 minus 10 percent. The complexity effect here is that when many heat pumps in an area ask for more power at the same time, the voltage starts dropping. The control algorithm in the heat pump still needs the same amount, so the consequence of Ohm´s law means that the current increases to compensate for the lower voltage. Heat pumps are normally three phase units, so depending on design, the fuse on the most heavily charged phase will eventually blow. A more insidious case is when the local transformer trips because of overcurrent. Then all houses in the area will be experiencing a brownout or blackout. The system cannot easily return to full operation, because there will be an accumulated deficit when the supplier tries to put the transformer on line, because the actual temperature in the houses will be lower, much the same way that the integrative part parameter in a PID loop can cause a delay of return into the area of control authority, the windup phenomenon. Many boundary conditions of complex systems display this effect of "stickiness", which makes restarting the systems more difficult. By power design engineers this is a well known effect, called "cold load pickup".

A network of heat pumps shows aspects of integrative complexity, especially when they are connected to the same electrical grid and when they represent a significant portion of the heat generation in that area.

Example 5.14 *Automated assembly line*

The automated assembly line is arguably the most common manufacturing process in the world. It is difficult to make a general description of this process. Therefore, this section will be devoted to the principles necessary to follow in order to automate an assembly operation.

Many assembly operations are semi-automatic, because the control theory developed is much more complete for continuous processes than for discrete event processes. The house assembly example given in Example 5.10 is such a semi-automatic manufacturing process. The fundamental reason for the lack of unified control theory for discrete manufacturing is that the mathematical representations of continuous processes are differential equations which provide answers at every parameter set, whereas the equations that describe discrete manufacturing have diophantine solutions. This is a major source of complexity, since many simple diophantine equations still have not been solved. A diophantine solution is a solution with only allows integers as solutions, e.g. the solutions where you produce 12.67 trucks are discarded. It is either 12 or 13 trucks.

However, it is not necessary to investigate every product that a particular manufacturing process is able to produce, only the product we want to make. The way to work out the methodology is to divide the assembly line into sequential steps, and then determine the conditions on which the product is allowed to proceed to the next step. Practically, this is done by registering all I/O points pertinent to the manufacturing as fast as possible.

Ideally, most manufacturing state transitions are made when the 50Hz power cycle makes a zero transition, which means that a control system cycle time of less than 10 milliseconds is likely to catch all normal events. By comparing the relevant I/O settings with a go/no-go vector for each assembly step, the process can be automated without necessarily solving the equations that describes it.

The manufacturing process is greatly simplified if a readable identity on the product that is being produced can be provided. In this way automated inspection and various quality measuring systems can be integrated into the assembly process. Also, production data can be stored per product, which makes it possible to register where the products are delivered. This tracking is especially important for the manufacturing of pharmaceuticals.

The complexity in the assembly process is related to completeness. If the process has states where the transition to the next state is a “dead end”, resources will be committed but blocked. For example, if there are a number of transport robots in a plant, the production pattern can result in oversupply of transport robots in one of the areas of the plant and a shortage in another.

The Scania factory in Oskarshamn is another example of a manufacturing operation which is guided by a product “recipe”. When a truck is sold, the design software will make a complete production recipe that contains all necessary information needed to build the truck. Subsequently, the recipe will be partitioned into subrecipes for the different production plants. The whole system will come together again at final assembly and test. When the tact on-line measurement system was first introduced in 1995, it encompassed 150 000 I/O points, each registered 20 times per minute.

Example 5.15 *The automated delivery warehouse*

This example comes from a real life system, which incidentally, supplies control equipment. Orders are received and checked against available stock and sent electronically as soon as it is technically validated (against customer credit limits and that the product combination results in a workable system). The order is then sent to available workers in the warehouse who will place the products into a sack that hangs on a conveyor system. When the sack passes a spot where a product is located, a lamp lights up and the sack stops in front of the location. A warehouse worker places the ordered units into the sack, which then moves on to the next stop according to an optimal route until the whole order is complete. At the same time, the dimensional information of every item is passed on to a system that calculates the optimum size of the carton. Another system fetches a flat (unfolded) carton, folds it correctly and provides it to the shipping clerk who takes the products from the sack and puts them in the carton. Every bar code on the item is re-read and a printer will provide a packing slip that goes into the box and a label on the box which then is transferred directly to a shipper in a loading bay. When the delivery truck is full, another backs into its place for new orders.

The modern self-service grocery store operates using a similar principle. Upon checkout, the shopper puts the bag on a scale. The system knows the weights of the products sold, and can alert the shop attendant if there is a suspicious weight differential. This is another example of using boundary conditions for control.

The automated warehouse process is, in itself, not complex. The complexity lies in the number of combinations that has to be considered when programming the functions. Each order requires that all products in it are available, so the shipment can be fulfilled completely. Otherwise a partial shipment is created. It would be possible to construct a system that can handle partial orders, but the number of combinations that has to be programmed would be very large. In order to get a working system at reasonable cost, a certain oversupply of inventory had to be accepted. This system condition was expressed by the warehouse system designer Tim Hohmann with the words: “If you haven´t got it, you cannot sell it”.

The solution here is similar to the situation where an intranet is used for communication between control equipment. This application requires determinism, and a TCP/IP protocol on IEEE 802.4 is not deterministic. The IEEE 802.4 is a standard for information processing systems. Part 4 of the standard covers token-passing bus access methods and physical layer specifications. However, by overdimensioning the communication speed and restrict the ability of the connected devices to send unlimited amounts of data ("jabbering"), the system will work in real life even if there are theoretical (complex) cases where it will not function.

The warehouse is a common system but it is poorly understood from a control standpoint. A typical layout is shown in Figure 5.5. Many users have ERP (Enterprise Resource Planning) systems that determine reasonable stock levels based on the history of previous orders. However, there is often the assumption that order sizes are following a normal (Gaussian) statistical distribution curve. The normal distribution is described by Hazlewinkel (2001). This may not be correct, and reliable statistics are dependent on a significant number of samples (in this case orders) that may not be available for low volume items. If another modelling assumption is used, such as the probability of being able to fulfil a particular order by current stock, another pattern may emerge. Therefore software and models for big distribution centres may not be suitable for small ones. Also, spare part centres who have many part number but few orders for certain items may have a different distribution.

In reality, overstocking is a common way of handling complexity. If the amount of customer orders is too limited, then there is much less risk of having an insufficient number of units available in stock to meet a particular order.

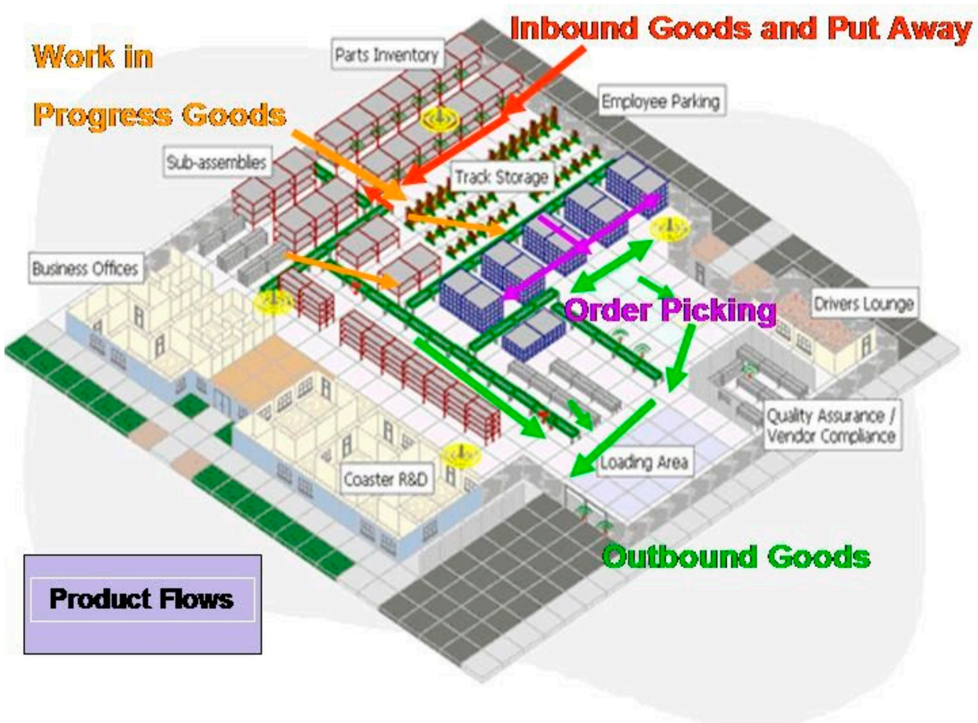

Figure 5.5 Warehouse layout. Picture from DCM projects.

There is ongoing work in this area using industrial AI (AI=Artificial Intelligence). The inventory levels can be fine tuned by applying AI techniques to analyze the shipment and order histories. Wilkins (2018) discusses these applications.

The technique is used in financial systems, where assets like stocks are traded by computers (program trading). The system assumes that the ability to fulfill orders is unlimited (which it is not), and if there are no buyer program active, a value of a particular asset can fall to extremely low levels, essentially zero. In order to prevent this, system clearing houses suspend trading if the price change gradient is too steep. The application complexity makes this system fail. In economics, failure is an option, because some systems are constructed unstable. If the sum of all current transactions never evens out, the system will gradually drift to a boundary condition and stay there. Some parameters, like debt, must then be reset before the systems can be restarted again.

Example 5.16 *Retention control in a paper machine*

> A paper machine puts pulp onto a continuous wire mesh where it forms a web of lignin fiber, which is the main compound in paper. Water drops through the wire mesh and the amount of dry matter that is withheld is called retention.
>
> The fibers that do not stick on the web are recycled for another try. The issue with this process is that it is recursive and can cause accumulation of bacteria, since the pulp contains monosaccharides and the process is conducted at room temperature, rather ideal conditions for bacterial growth.
>
> In order to control bacterial growth, biocides are used. However, successive generations of bacteria can adapt and render any particular biocide ineffective. Therefore, bacteria are killed off by a shock treatment with temporary high doses and different biocides, which are used alternatively in order to prevent bacteria to adapt.
>
> This type of process is complex in a special way, in that conditions vary with the time a process has been in operation, and specifically with how many iterations a particular material has been processed. Any accumulation above zero will eventually reach 100%, and in order to break this circle, a constant fraction of new product must be introduced in order to keep the process from reaching a boundary condition in the form of a process threatening concentration.

This application is a typical example of recursive complexity, as the complexity increases when retention is low.

Example 5.17 *Paper recycling*

It is common to recycle paper to produce new paper, sometimes of inferior quality. However, with each recycling, many fibers become shorter by the mechanical process (grinding) and when too many fibers become too short, they will not form a sustainable web. This effect can be counteracted by adding gluelike components but this is not as environmentally friendly as the recycling process is intended to be. Therefore, the complexity inherent in this type of processes may indicate that other methods, such as controlled combustion, in order to retrieve energy, may be the best option for products that has passed through "too many" recycling iterations.

Example 5.18 *Non-equilibrium systems*

There is a growing concern that some systems that we rely on are not stable in the long term. They are based on a finite amount of resources available that eventually will be successively exhausted and the problem is solved by abandoning the old system and restart the process again. From a control standpoint, the situation is analogous to having an integral action in a PID controller, which is increasing all the time and the problem is solved by periodically setting this term back to zero, like an anti-windup function.

A common situation that most people can relate to is the dependence of the Microsoft Windows operating system. The system is so large and complex that there are many software functions that use resources of the operating system, like file handlers, stack memory, software interrupts, etc.

If a certain software function is used frequently and if it fails to "return" resources (like memory used in stacks) to the operating system so it can be used by other programs one will eventually run out of a particular resource, and the system has to be restarted in order to release the blocked resources. If the operating system were not periodically restarted, it would degrade because of some resource being exhausted. The only way to reasonably avoid this to happen is to start a limited number of programs and not change the number or close/restart the threads. If all programs were written so that they returned the resources used from the operating system when the thread is closed, the problem would not be so difficult. Unfortunately, it is still reasonably easy to write malware that exhaust the system resources by creating more open file (handles) that the system can cope with. An even more sinister method is designing malware that exhaust the computing resources of the system. One such example is using the encryption algorithm for passwords. It is advocated to have long and complicated passwords, but if the entry is not limited, malware can preoccupy the encryption algorithm by entering a 10000 character long password which will block the processor for an extended period.

This situation is closely related to complexity, because the problem we try to solve is too complex to completely model, so the control is an approximation which handles the situation for the time being. However, if the process is operated for too long, then the process will eventually stick to a boundary condition. If the system is not reset, then it will be glued to the boundary indefinitely. The fact that big systems, in this case electrical power systems, have particular vulnerabilities has been studied, for example by Johansson (2007).

Another system that may be a non-equilibrium system is the way payments are made within the group of countries that have adopted the Euro. Payments between Euro countries are no longer administered between countries but cleared through the European central bank. This system is called TARGET2 and as long as there are persistent deficits in some countries and surpluses in other the system accumulates in ever increasing credit and debit accounts. It is like issuing a credit card with no limit and no definite time on when the bills are due. The resulting effect is called *TARGET2 imbalances* and the system works for now but we do not know where the boundary condition is until we reach it.

The situation can be illustrated with the Figure 4.1 of the polygon in Chapter 4, only that the polygon is not closed. The imbalance represents an area which is "open" not constrained by a boundary condition.

The tradeoffs and difficulties in controlling the volume of Euros is described in the book "The Euro and the Battle of Ideas" by Brunnemeier et al. (2016).

Example 5.19 *Extrusion of plastics*

In the 1980´s the author was contacted by Tetra Pak Research & Development to help with a control problem. The company was trying to develop a material with insulating properties, essentially a plastics foam. The foam was called XPS (eXpanded Poly Styrene). Air was injected into an extruder, and the plastics was extruded through a narrow but wide lid. The thickness of the aperture was controlled precisely with the help of 65 thermobolts. The problem was that the foam was not smooth, since the air bubbles were bursting like popcorn.

It was quickly determined that this was a control problem. The thermal controllers had an accuracy of plus minus 3 degrees centigrade, whereas automatic control which would prevent the bubbles to burst had to be 10 times better. The problem was that the mechanical device was functionally complex, one bolt setpoint would affect the two bolts next to it, so there was no way the thermobolt PID controller software could be implemented separated in different processors. A calculation was made and we found that we had a maximum of

86 milliseconds to perform the required calculations and send the actuator control signals to 64 of the 65 thermobolts (thermobolt 65 at the edge was used for reference). Forty years ago, this was a challenge when the processor speed of a 16 bit processor was 1 MHz. The problem was solved by building a processor for this specific purpose. We could not use dynamic RAM, because it had wait states, so we used static RAM. The PID controller had to be written in Assembly language.

It can be argued that this project has no relevance now two decades into the new millennium. However, it may. The reason is that processing speed is never infinite, and there are control applications that still are dependent of knowledge of how processors work, especially when RISC Processors are used (RISC = processors with reduced instruction sets). The reason is whereas most industry devices are constructed with processors that use symmetric Assembly, most commercial processors use Intel processors that uses asymmetrical Assembly. Asymmetrical assembly was important 40 years ago, when conserving memory was a priority. However, it is easier to write reentrant code in symmetrical Assembly and this is important when the operating system has to make many context switches in a short period of time.

Another recent and important consideration is when systems have to be made resilient against hacker attacks. It is important that simple and reliable structures are implemented so that control systems for vital functions in society are protected against malware. One such method is to use multi-core processors where one processor is continually calculating checksums on the application. If corruption is detected, a rapid context switch to a non-corrupted version is made.

The Tetra Pak application was the first where the author became aware of the issues of functional complexity. As a result, a SCADA package was developed and later deployed in a plant which produced 10 000 tons of plastic bags annually.

Examples Summary

The different categories that have been identified through the examples should derive from common technical properties. For functional complexity, the polygon depicting the boundaries can be complicated, extended if production capacity is increased, for example by adding shifts. If some setpoints are not achievable, it could be shaped like a donut, as in the Example 4.3 with the aerator compressors in a wastewater plant.

For operational complexity, the shape of the polygon can expand and contract with time, and integrative complexity represents that more and more limit faces are added and if too many are added that constrict each other, the control space can become a singularity and the polygon can deteriorate to disjunct shapes, separated from each other.

Recursive complexity, finally, means that the faces of the polygon becomes successively blurred, like adding a deadband to every variable, so that the area of control authority can no longer be accurately determined.

The structures are further discussed in Chapter 6.

6. Suggestions for generic structures

Based on the observations from real life production processes, we will put forward a hypothesis based on the in natural sciences common deductive method, described by Karl Popper (1959). The hypothesis is that the limits of practical control solutions developed for the applications analyze in Chapter 5 can be described in a unified way as modifying data and providing outputs by applying a control scheme. The method to analyzed boundaries is qualitative and not quantitative. The purpose is to find structures that can be used.

The generic equation for this type of operation is simply

O = F X **I**

where I is the input vector, O is the output vector and F the algorithm that describes how the inputs are transformed by the process to form the outputs, the function we call the ***function state matrix***.

Based on the various examples in Chapter 5, it is proposed that the different production processes can be classified according to their type, grouped into categories and analyzed using common tools. However, this model describes how automated control is made ***in real life***, where the functions supported by the systems are the ones that are deemed to be profitable to control, the ones that technology is able to control or just those that has been effectively implemented in practice.

However, it is important to note, that this model can only be used to ***analyze*** the complexity of the control system model. The function state matrix is linear, whereas many control system models are nonlinear. The concept is ***related*** to control theory models, because if it is multivariate and linear, it will become a transfer function matrix, known from control theory and in the single variable linear case it will be a transfer function. However, in most cases it can only be used to complement the control design, not replace it.

This means that the function state matrix may not be complete, or it may even contain too many conditions, so that certain actions could be ambiguous, i.e. leading to unpredictable outputs.

A system with too many conditions may even oscillate between two states, depending on the frequency of condition evaluation in the software.

Every hypothetical assumption contains scientific uncertainty, and if there are industrial processes that will not let themselves be described by the tools suggested, then the hypothesis is falsified. However, the number of actuators, measuring devices and controllers are finite numerable and the bulk of all processes are constructed from a limited number of unitary processes and therefore the combinations can be predicted. A lot of work has been done to describe unitary processes (see Chapter 8.2) and even provide automatic programming of those.

However, this thesis will put forward the presumption that the unitary process description will not be sufficient to avoid complexity resulting from combinations of unitary processes. Indeed, depending on the total size of the process, a functionally combined system will place additional requirements on the unitary controllers (e.g. the refinery Example 5.3).

Combining many simple processes can itself lead to complexity. This mechanism is called ***emergence.*** The process of generating complex structures from simple assumptions has been widely studied, by Radestock and Eisenbach (1996) and Holland (1997).

By analyzing various production processes, the hypothesis is that the aspect of complexity in automation systems will fall in one of the four categories below:

- Functional complexity,
- Operational complexity,
- Integrative complexity,
- Recursive complexity.

The method will be illustrated using a simple example. Let us use the common faucet, described in Chapter 1.3. We agree that it does not contain any complexity and it can be described with the following equation:

$$\mathrm{F} = a * p$$

where F= flow, p = water pressure and a = the angle by which the faucet is turned.

The permissible flow is also determined by the dimensions of the faucet pipe, the pressure should be in the range [0, 500 kPa] and a = [0, π/4] radians.

If we want to calculate the range of flow in a building with n faucets, then the function state description would be:

$$\mathrm{F}(n) = \sum_{i=1}^{i=n} a_i * p_i \text{ where } i = 1..n$$

There is still no complexity.

In a tall building the water pressure may depend on which floor a particular faucet is placed, which means that $p = p(h)$ where h is the height.

When this constraint is added, ***complexity*** is introduced. In order to calculate the combined flow, one has to know the position of the faucet in relation to the supply.

In a skyscraper, a pressure equalizer may be installed in order to maintain faucet pressure. This would be a pump that brings water to a storage tank high in the building. If this is placed at level H, then

$\mathbb{P} = p\ (H\text{-}h)$

Since the pump and tank capacities are limited, this will most likely add constraints on how much flow that can be extracted in a combination of faucet settings.

When the first high-rise buildings were built in New York, it was custom to include a water tank in the highest part of the building.

However, pressure may not be available at all times. During the Second World War, various parts of Stockholm had only hot water one day per week. In areas with water shortages, access to water may be limited by rationing at certain times.

This means that $p = p(t)$ where t is the time of day.

To calculate the possible cumulative flow rates permissible, one has to consider the time dependency, which we define as ***operational complexity***.

It is not so easy to design an example of integrative complexity based on the faucet example, so let us consider a heat flow meter. These are used in, for example, the distribution of municipal heat (district heating), which is distributed via isolated pipes and heat exchangers at the delivery points.

Consumption varies with the withdrawal of hot water and the need for keeping radiators warm. The flow and temperature in the distribution system can be varied, but within certain capacity constraints. If more and more consumers are added to a network, integrative complexity will occur when they exceed a certain number. Before this point, the system is characterized by operational complexity. Operational complexity occurs as a result of the time varying aspect of the consumption pattern, but the integrative complexity manifests itself as a reduction of the space that can be controlled, the control authority. When the control authority space shrinks to zero, the requested amount of heat cannot be reached and the temperature setpoints have to be adjusted if the system is to be brought back under control.

If another heat distribution system is designed with a network of heat pumps, having electrical energy sources, then the situation becomes different. The distribution of electrical energy has a hard limit, which is the transformer capacity and if this is exceeded, the transformer will trip. No consumer in the network will get any energy until service is resumed. In order to maintain control, the temperature setpoints will have to be altered ***prior*** to the transformer supply limit has been reached.

This makes such networks a more attractive candidate for calculations of integrative complexity, especially if the various heat pump setpoints can be altered dynamically over a wireless network. Such tests have been done in Sunnersta north of Uppsala (2017).

Two particular aspects can be mentioned in this context.

Firstly, both systems experience a remanence similar to the anti-windup of the integrative part of a PID controller. If the setpoint cannot be reached, the "deficit" in energy requirement will accumulate. In a water based distribution system it can lead to an "overshoot" in demand after the system delivery capacity has been restored. In the case of electrical energy, it can cause the substation to trip again just after service has been resumed. This effect is called "cold load pickup". The network services will therefore have to be reestablished in "phases".

This effect is well known in the manufacturing industry. When a large production line is started, all motors are not started at once. Instead, they are started sequentially, "along" the production line, in order to cope with the inertia in the system.

It should be noted that the function state matrix description does not differentiate between continuous and discontinuous processes. This is because the model can contain digital variables as well as analog ones. The user of the model just fills out the matrix elements as they are known and when a sufficient number of relations between inputs and outputs can be identified, a space of control authority will emerge. Space outside the multidimensional polygon is where complexity rules.

6.1 Functional complexity

Definition:

A system will display functional complexity if the relations between inputs and outputs are complex.

One common example would be a system where many inputs can affect the outputs, i.e. where the input vector is large. This complexity appears as a result when the functional state matrix equations describing the boundaries of process dynamics are not mathematically satisfied, e.g. there are more parameters than equations that can be used to resolve the parameters.

A purist would claim that such a situation would be uncontrollable and hence require further study before being automated. However, in a more practical world we try to control insufficiently researched processes all the time, in particular biological processes, because the methods or processes used appear to have inherent stability despite the fact that a fully resolved process model is not available.

Functional complexity can also result from too many conditions, for example when redundancy of components is introduced. If there are two identical sensors where one fails and the secondary sensor has a different value, this will be considered a step change by the automation system even if the underlying process is unchanged.

The field is large. For example, consider the relationship between the temperature in one place vis-a-vis the solar radiation intensity at the same spot. We know there is a relationship, but the equation that will describe this function between solar radiation in W/m^2 and the ambient temperature at a particular point is surely complex.

Solar energy is a good example of one input and many outputs, connected through a complex dynamic model.

Another area, where complexity is a factor, is where one variable is tightly controlled at the expense of losing control of other variables. This can be exemplified by a speed bump build in order to force vehicle drivers to reduce speed. Speed will be reduced, but since the vehicles will decelerate before the bump and accelerate after the bump the engine efficiency will be reduced and result in higher CO_2 emissions (for engines based on fossil fuels). Even for electrically powered vehicles, driving costs will increase. The braking followed by acceleration requires more energy. Furthermore, the wheel suspensions are designed for a number of contractions/expansions which is statistically finite.

Industrial production facilities are full of such trade-offs: in a rolling mill, if energy is saved in the heating oven one may have to pay for this with more product scaling in the rolling process.

Functional complexity is the area that has been best understood. There are still new complex machinery being constructed, such as machines that can peel shrimps or fry pancakes, but in many cases the increase in complexity is not matched by a corresponding increase in productivity. One example is the filling speed in newly developed filling machines from Tetra Pak. The speed offered by new machine models has tapered off because a doubling of the filling speed is counteracted by the fact that such a fast machine will more than double the functional complexity.

The two main areas that are the focus of functional complexity are integrated manufacturing and robotics.

In integrated manufacturing it is now possible to completely automate the production of certain products. One example is three-dimensional electronics, using 3D-MID technology (MID = Molded Interconnected Devices). A production recipe that completely describes the manufacturing steps for a product is segmented and converted into subrecipes, which are downloaded to a group of interconnected machines. The machine group will then manufacture and test the product.

One may observe that fully automatic manufacturing is frequently possible, whereas semi-automatic (where some operations are performed by humans) is exceedingly difficult. The reason is that it is possible to construct a function state matrix based on a model with a finite number of states for a machine, but to do so completely for a human is probably impossible. The complexity effects relating to the corresponding manufacturing organization has been studied at Volvo Cars by Pasche and Persson (2008).

The author was engaged in solving a problem arising from semi automation, debugging a process of recycling solvent chemicals at a pharmaceutical plant in Mölndal, Sweden. The system actuators were controlled via a local intranet. The filling of a tank was tested, but during the test a technician disconnected a switch in the intranet (by pulling out a network cable). The result was that whereas the ***start fill*** command could be sent, the ***stop pump*** command could not. There was a nervous moment in the testing process until the emergency power breaker was used. If the filling had been done fully automatic with a local controller, this type of error could not happen. Later, the problem was solved by adding a token to the network protocol so the process would be put in a safe state if communications were somehow interrupted.

The second area, robotics, is an area of functional complexity that is rapidly expanding. There are many new types of robots that cuts grass, cleans windows or collects dust. New robot types are developed for working under dangerous conditions, such as in mining operations or for the demolition of old buildings. The author was involved in 2014 to develop communications to a subsea robot used to recover valuable cargo (copper) from a sunken ship off the coast of Mozambique. New types of robotic functions in various operations are also being developed, such as the removal of insects on edible herbs distributed in pots with puffs of air.

One major area which stands out is the development of automatic vehicle control, allowing the vehicle to move on roads autonomously. These controllers are already safer than human drivers but the few number of accidents caused by automated vehicles get much more attention than the fact that human errors causes millions of accidents every year. It seems that we have less patience/oversight with errors committed by automatons.

There are miniature robots that can perform micro surgery and the merger of robot miniaturization, artificial intelligence and pattern recognition have made some really sinister military applications possible ("killbots"). The idea of autonomous robots fighting wars was first proposed in the 1970´s by Vaughn Bodé in the comic book "Junkwaffel". Here understanding of functional complexity may be a condition for survival.

The last area we will cover here is the new area of collaborative robots (cobots). Traditionally, robots have been placed in cells and if a human entered the cell, the robot would stop its operation. There are robotic functions that perform cleaning cycles at flushing toilets or help to wash elderly and to complement functions for people with disabilities but the biggest area is the execution of simple tasks. Already motors are transported by cobots at the Volvo Car motor assembly line in Skövde, Sweden and are used to move trays of hard metal pieces between hardening ovens at Sandvik Coromant in Gimo, Sweden. The cobots move slowly and if a human stands in its way for too long it will give off an annoying noise. The most common version most people will see is probably a street delivery cobot.

Delivery drones are also to be considered as an example of robotic functions. There are already many installations where drones operate daily delivery schedules between two locations. Tests have been conducted in the Singapore harbour, where deliveries to ships moored outside have been conducted by drones. There are also test of drone deliveries across Elliðárvogur, the inlet that separates Reykjavik from its eastern suburbs. In Africa, a drone service connects Mwanza on the shores of Lake Victoria with a clinic on Ukerewe Island. The service sends samples from the patients in the clinic for laboratory analysis and the trip takes 40 minutes as compared with 6 hours by ferry and road.

6.2 Operational complexity

Definition:

A system will display operational complexity if the function describing the relation between inputs and outputs varies over time.

Operational complexity is found in production processes where the setpoints and other parameters changes over time, i.e. the function state matrix is not **F**, but ***F(t)***.

Operational complexity is very common, because most equipment changes its characteristics as a result of wear and tear. However, the complexity is observed only if it affects outputs in a way that has practical consequences. Many processes are designed so that they can absorb changes in inputs as long as they are within a certain range, a bandwidth. This complexity is more common when the desired output setpoint is time dependent, such as need for varying outputs. A common problem is when the setpoint is outside the control authority, e.g. when a higher output is required than the system can provide. For example, a valve can only be 100% open.

One big area of growing importance for operational complexity is when big processes become even more integrated by sharing one or more common resources. This could be steam or water in an integrated chemical complex or the pipe grid in a large refinery tank farm. This area has been so much in focus that it has been given its own name, CPSOS. CPSOS stands for Cyber-Physical Systems of Systems and has been studied by Engell (2017) and by Stepanovits (2017).

It is important in this context to differentiate between variables that has time dependencies, such as inputs, and operational complexity. Operational complexity is considered when the input/output relationships, the function state matrix, themselves change over time. There are many such systems that we have been accustomed to use, such as in transportation, heat distribution and other grids.

If a road is blocked, traffic will flow some other way if possible, and if some resource is missing, the operation in which it appears is affected. If there is no red paint no red products can be made. This is an area of great economic interest since advances in understanding operational complexity has made it easier to organize automated production better than to drive functional complexity limited operations closer to their boundary conditions.

At first, it can be considered somewhat unorthodox for an automation engineer to study factors like pricing of products or raw materials. However, when analyzing operational complexity, parameters other than those that are emanating from physical measurements have sometimes great impact. At a pulp mill in central Sweden, the company sold the condensed water from the steam serving the pulp boiler to be used as heat in the municipal grid. During a cold winter period the boiler was kept running during the annual Christmas shutdown because the price of municipal heat was higher than the corresponding internal price of the energy for pulp production at the mill.

Taxes can also affect operational decisions, if the tax on municipal heat is increased, it can lead to reduction in electricity generated by cogeneration, which can lead to regional shortages in electrical power availability.

These external constraints affect process design in many areas. This is especially common in raw and intermediate material production. At many upstream production sites, where crude oil is extracted, the oil is accompanied by natural gas. If the extraction site is in an area where gas recovery automation, for example LNG production, is deemed too expensive, the gas will be flared off into the atmosphere. One such area is in the Niger delta in Nigeria (Olsson, 2015, Ch. 11).

For global corporations who have several manufacturing units, which all can produce the same product, exchange rates may influence the monthly planning of how the volume of units should be partitioned between the sites.

The main effect of operational complexity on plant design may well be the fact that the cost of removing bottlenecks in the physical plant varies considerably depending on the level of operational complexity. One such example is the delivery of standard chemicals, such as sodium hydroxide, at Akzo Nobel, Sweden. The hydroxide is used together with acid to disinfect tanks after producing batches of liquid foodstuffs, such as milk. By changing the operational design of the supply chain the physical production cycles can be smoothened out and a careful analysis of operational complexity leads to insights affecting the process design.

Since functional complexity is well understood, more processes have been developed to operate closer to their respective optima than in operational complexity. Here, there are large possibilities to obtain better functionality. It is often easier to organize production closer to plan than to try to get machines or people to work faster.

6.3 Integrative complexity

Definition:

A system displays integrative complexity if the range of control authority changes depending on the number of systems interconnected.

Integrative complexity is more common in supply systems and more rare in continuous processes. It occurs when the output of one system is fed back or accumulated and the result changes inputs to other systems that are interconnected in a joint configuration and the combined effect results in change of control. A common result is changes in boundary conditions, resulting in a successive reduction of control authority.

Often the conditions under which the process can be effectively controlled are reduced (effectively locking the system state to a boundary).

Integrative complexity is insidious in nature and can create large unanticipated disturbances. Many systems grow organically and it is not obvious when an overall system passes a size threshold, in other words when the sum of all possible rare occurrences becomes so significant that it will affect the system´s overall performance.

It has been shown by Holland (1997) that highly complex systems can appear as a result of combinations of a large number of simple systems. Suggestions of how to handle this has also been proposed by Sha (2001).

The emergence of many large systems of interconnected devices through IoT (Internet-of-Things) is a growing area of interest for the study of integrative complexity.

If a large number of units use the same software, for example from open software sources, then the units may all act in such a coordinated fashion that bigger disturbances are caused to the interconnected system they form than these are designed to handle.

Emerging areas of integrative complexity are when publish/subscribe access to the Internet is used and ad hoc subnetworks are created. Most common participants are computers and/or smartphones. In both cases, both beneficial and less desired effects can result. Computer owners can offer processing speed in order to help solving meteorological or astronomical equations that require very large processing power, but processing power can also be hijacked for the illicit mining of cryptocurrencies. Smartphones can be used to locate the closest available doctor in case of an emergency or to be used to create flash mobs.

One large emerging area is the dynamic pooling and reallocation of resources, for example in transportation or in optimization of storage, such as reported by Gillblad and Johansson (2016).

There is also great potential in connecting production processes of intermediate products with the storage facilities of customers, thus extending the capacity boundaries for the producing units.

6.4 Recursive (iterative) complexity

Definition:

> ***A system displays recursive (iterative) complexity if the range of control authority changes depending on the number of times the output is recycled through the inputs until the process is completed.***

This type of complexity is not common, but there are a number of production processes that are designed in such a way that some of the output is recycled and fed back a number of times until the process is deemed complete. Sometimes this can only be done a maximum number of times after which the process function no longer works as intended.

However, there are many control situations where iteration can cause complexity issues. All digital calculations have rounding errors, and if a recursive algorithm is used to calculate a parameter, its value successively degrades. The situation is similar to changing money several times between two currencies. If this operation is performed repeatedly, let us say n times, over a certain value of n, depending on the size of value destruction in each transaction, the original capital will have dissipated.

There are special areas where the analysis of recursive complexity is absolutely essential, such as the water recycling in the resource constrained confined space of a space station.

It is well known that certain recursive processes can lead to adverse effects, such as in inbreeding and mad cow disease. However, there is growing evidence that cyclic accumulation can also occur in processes controlled by automation.

More and more process industries are installing in-process recirculation systems, which will mean more systems displaying recursive complexity. One such example is the remelting of steel scales produced in the rolling mill process, discussed in Example 5.8.

The author was first aware of this type complexity when programming the temperature in order to control a steel making process at SSAB in Luleå. The programming was done in Assembly language and the temperature probe was a thermocouple placed on a rod that was gradually moved toward the steel surface in the furnace. In the end, the probe was melted by the steel and the program function was to create the temperature curve so that the end temperature was estimated. If the process iterated too many times during the insertion, accuracy deteriorated. There was an optimum number of iterations for a given accuracy.

7. A practical approach to handle complexity in industrial production systems

There are problems validating the hypothesis presented in this thesis. Complexity is a growing topic because it appears more commonly in large systems and the number of large systems is indeed increasing. However, most owners of large systems are unwilling to subject their systems to introduced disturbances in order to test the system resilience or to analyze the deficiencies/vulnerabilities in the function transfer matrix. There is a lack of test facilities, so the most common alternative is simulation.

A disturbance that leads to a breakdown of functions is also an event that puts pressure on the process owner to get the system up and running again, and the restart process may obscure data necessary to be able to do a good analysis of the root causes.

Analysis can be improved if the process owner keeps good records of noncompliant situations. There are methods for this, like Kanban, but most process owners are unwilling to share this information outside their own company. A description of Kanban has been provided by Anderson (2010).

7.1 Consequences

This thesis argues that complexity is a major cause for industrial control projects being less successful than expected. There is ample evidence that the percentage of industrial control projects that fail to meet their targets is high. For this reason, a special standard, ISA-95, has been developed. However, its recommendation is over a hundred pages. An overview is provided by Scholten (2007). There is a need for more scalable methods.

Failure to meet expectations can take many forms; there could be delays in implementation, cost overruns or failure to produce planned volumes of the desired quality. As society becomes more and more dependent on control systems, and the fact that these systems are installed in ever increasing numbers mean that the problems will not go away. Also, there is a suspicion that some spectacular failures of control systems could be attributed to a failure to program correct handling of certain input combinations. Typical examples are violent swings in stock market share prices caused by automated trading and a shortage of automatic buyers, power grid brownouts and large failures of power plants resulting from adverse weather conditions or solar flares.

To put more effort into the design of operational states in every individual system may not be an option, since the number of control systems will exceed the number of humans on earth within an immediate future. The traditional way of designing control systems has been to use experience from older systems already in operation. However, this method may not be sufficient in a connected future where more variables affect each system and these variables have not been considered when the older system was designed.

What will happen? We do not know, because there is no way to check a system against all possible combinations of inputs. Nobody is really responsible for the overall consequences. This is a true trial and error situation where most people whistle in the dark. The real situation could be even worse. The fact that complexity sets a limit to how much money can be spent on programming, testing all possible outcomes constrains the possibility to build more complex system than we have today and thus puts a brake on development of new capabilities.

If a particular system has a model developed for it, then there are methods emerging for automated testing of the model, such as proposed by Abbate et al. (2010). Another example is the development of "digital twins" for people. Then treatment options could be analyzed against the digital model before the patient is administered treatment.

If we do not get a better grasp on the causes and reasonable actions to reduce complexity, system size may eventually put a limit to what can be done.

Example 7.1 *A master/slave industrial network*

> Some older networks require all nodes to be on-line before the polling will work. This means that reliability decreases with system size.
>
> There are IoT type systems that have been installed where a server polls data from nodes. One example is collection of air quality data from air filters. The collection was done via 4G modems, and initially a program was made to collect these data which had a polling cycle of about 7 minutes. Then more nodes were added and the system is now comprised out of 43 nodes. The polling

cycle is now about 5 hours. When the system will reach just over 200 nodes, then a full day (24 hours) will be necessary to collect all data, and the system can grow no further. The problem here is that when the system is put into operation, it is difficult to change it. Even if this is done, it is difficult to homologate the new data with the old data collected.

This illustrates the need to design node software in such a way that it becomes ***scalable***. This is a direct consequence of integrative complexity. The (eventual) total size of the system can affect how an individual node has to be programmed.

This is especially relevant when it comes to designing systems of IIoT type (IIoT = Industrial Internet-of-Things). A decision has to be made on how to split functions between the cloud/server and the node (the edge computer). Edge functionality can range from fully dependent, like remote boot (called a ”pixie” boot for Intel architectures) to fully autonomous, autarchic edge nodes. Autarchic nodes do not respond to polling (pinging) but participate only if called from a particular request handling server, a broker. Modelling for this new type of system architecture is further discussed in Chapter 8.3.

Furthermore, if the understanding does not increase, then we may take risks when we connect various systems that we are not aware of. This is particularly true when people have to disregard certain possible outcomes in order to meet deadlines. The most common/desired systems states may be taken care of programmatically, but other possible states may be undefined, or as this thesis states, complex.

Sometimes, a polling cycle with a maximum delay is imperative. The author was working with the Danish Rail Authority in 2013 to create an internal network in a locomotive that connected several units, which in turn read the position of the locomotive. This was because different rail networks have different identification systems, and the objective was to be able to use the same locomotive across borders. By setting several connected receivers in the same locomotive this was achieved. Obviously, the overall condition was that the rail networks had the same gauge.

7.2 Method

When designing a new system, this thesis argues that there is a preferred method to analyze the control problem so that the system will be optimal with regard to functionality. The aim is to reduce complexity so that the system can be implemented towards desired performance.

The first step is to list the states the system is supposed to be in. The initial system states will be at the top level, and then individual subsystems can be further elaborated. In complexity research, it is recognized that the most common models for describing automatic control do not take complexity into account. Studies have been made on this subject by Hyötieniemi (2003). Integrative complexity has also been studied, resulting from combinations of simple structures, a mechanism called ***emergence***.

So far, most of the research has been focused on theoretical aspects of complexity, but in real life most systems are implemented despite the fact that they do not solve 100% of the control problem.

The approach proposed is a step-by-step method which can be applied to any process, continuous or non-continuous, in process industries or in manufacturing.

The advantage of this method is that it does not have to be complete. The user will start populating the matrix in steps, starting with the functional relationships that are analyzed to have the highest impact. Then more functions are added until the process has established a space within which control can be implemented. Likewise, all outputs do not have to be considered. One starts with the ones of interest and the corresponding input requirements are analyzed. Selecting a suitable control algorithm may also result in more constraints on control authority.

7.3 A step-by-step approach

The first step is to analyze the states the system can be in, at least the ones that can be identified.

A discrete event process may have consecutive steps, a batch process may be in states like start-up, running (in operation), in shutdown or alarm.

A good way of doing this is to analyze the states with the conditions for transition between the different states indicated.

In a common discrete event process, such as a complex assembly operation, it is a good idea to introduce a transfer vector, which contains all the measurement values (like position sensors) necessary in order for the operation to be allowed to proceed from one step to another. Since an assembly process like the one in Example 5.14 can contain well over 100 000 measured data points, computer control is the only way to administer such volumes. Also, this opens possibilities for event handling, like interruption of power. If the power outage is sub-second and the vectors unchanged, manufacturing may be allowed to proceed. If it is longer, positions affected by hydraulics or inertial effects could have changed and the correct action is to move the system into a safe starting position before operations are resumed.

It is important that automation programmers understand physical properties of equipment, such as inertia in order to write code that will function correctly.

Example 7.2 *A three-way two-port valve*

Simplistically, the valve can be in either of three states, left, right and in transition.

The input vector is simple,

$\mathbb{I} = p_i$, where i = [0,1] and p_0 = left position and p_1 = right position

The output $\mathbb{O}$ has in its simplest form three states,

$\mathbb{O}=o_j$, where j = [0,2] and o_0 = left, o_1 = right and o_2 = in transition

The functional matrix can be described as the chance the valve is in either of the three positions,

$\mathbb{F}= \mathbb{F}_k$, where $\mathbb{F}_0 = 0.50$, $\mathbb{F}_1 = 0.49$ and $\mathbb{F}_2 = 0.01$

The valve has to be in any of the known positions, so

$\sum_{k=0}^{2} F_k = 1$

However the matrix function is not complete, because the valve can be in more states (rare cases), for example

a. The seal can be damaged, so that product seeps in both directions,
b. The valve can be worn, so that a command will cause the supervisory system to report the valve as malfunctioning (time to perform action exceeded),
c. The valve can be leaking.

Other rare cases can easily be constructed, such as the effects of unit aging or lack of use (e.g. "sticking"). One example of sticking is the couplings between wagons in the Stockholm subway: some wagons have not been separated for so long the they have become "married", material sticking has made the couplings unseparable.

When automating a system comprised out of many valves, then the combined system will display integrative complexity.

If a simple automation program is written in order to control one valve, it may work for a long time without problems. However, if a system has thousands of valves, then the "simple" control program may not be adequate to control the combined system.

The conclusion is that the total size of the system affects the way a unitary controller functions. This puts a limit to how many "legacy" systems that can be connected to a supervisory network.

An easy method to identify the function matrix is to start from the end, since the desired output is known, then work backwards toward the start of the process. When the required number of steps has been identified, then the transition conditions can be determined. In most cases, these can be coupled to process output values. A tank mixing will be started when a level has been reached, or a transport conveyor will be started after a certain time has elapsed, etc.

The output vector can be constructed in this way. The next step is to determine which inputs affect the output. Then the input vector can be designed.

The last step is to determine the functional relationship between input and outputs, which forms a matrix over the states with a mathematical relationship between the corresponding inputs and outputs as the elements. In most cases, the matrix will be incomplete, because some relationships may not be known or, which is more often the case, deemed irrelevant.

It is not important to construct a complete matrix because the suggested method will work even if the process knowledge is incomplete. The author argues that complexity arises from the ***unknown*** parts of a problem, so one should use the available information. The parts that are fully understood are no longer complex.

The valve model is important, as there are many systems that are designed around large number of valves. One is refinery off-sites, where hundreds of large tanks are connected through a grid of pipes and remotely controlled valves. Another is replenishment of plastic granules in large extrusion plants, such as plastic bag production. In order to control such systems, assets are blocked through a "path" vector structure, and certain key assets, such as flow meters and scales, are often redundant. Complexity calculations of transportation grids are often difficult.

7.4 Validation of model

A first verification of this model approach was done using a system from Preem Petroleum, which implements measurements on some 500 diesel tanks, all connected to a single cloud service. The purpose of the validation was to calculate the aspect of complexity in such a large system.

Model:

O = **F** x **I**, where **O** = O_i ; i = 1..7

The seven output parameters are tank status, GPS coordinates, tank level, tank volume and battery status.

The input vector ***I*** contains 15 input parameters: max level, effective level, offset level, notice level, order level, alarm and empty level, tank type (physical form), dimensions, tolerance (filter type), position and sensor type.

A system report was presented on modelling of a data collection unit implemented for Preem Petroleum and presented by Gillblad and Johansson (2016).

7.5 Identification step

When a model is deemed sufficiently complete in order to be tested, the relationships in the transition matrix are investigated. By looking at the nature of the relationships, their type can be determined. The author argues that there are only four fundamental categories of complexity in automation systems, as it has only been possible to identify four types from the examples given. If a particular process cannot be characterized by one of these categories, it is probably because it should be further decomposed into smaller unitary processes.

Complexity can be reduced because there are many ready-made solutions for most control problems, and by applying solutions from, for example, a Modelica library (www.modelica.org) new system functionality can be designed without writing new code. Unitary process software can be applied in new combinations without debugging every unitary process yet another time.

7.6 Information formats

An essential component of a control system for a complex application is that the different software modules, constituting the complete solution, can exchange data with each other. Data has to be homologated and a common platform for description has to be agreed on.

One such common platform is XML (eXtended Markup Language) but a common platform is not sufficient. The different software pieces must also have access to the data when they are to perform their respective functions. A useful model for this is to regard the different functions as objects, where each object subscribes to the required inputs and publishes the outputs.

XML was proposed as a meta modeling tool at the ER2001 workshop by Fröhlich et al. (2001).

Work in this area has been done by Wagner (2002). There is as yet no agreement in the automation industry which platform that should be considered optimal, as there are significant commercial benefits for the company that can establish its model as a practical standard.

Early work in this area was directed to standardization of signal data formats, like OPC (Open Process Connect) and message services (for Microsoft Windows applications). New suggestions involve OPC UA (Universal Architecture) which enables cross platform support but there is still no system that handles all formats required to control all systems affected by complexity. Examples of recent data formats without standards are time series data, accumulator records and images.

There is a generalized version of the markup Language, UML (Universal Modelling Language). Recently, is has been developed into a version that can generate code, xUML. Open Modelica also has a language version, called OpenModelicaML.

Furthermore, a lot of standardization efforts have been made for various industries, mainly for security concerns, the railway industry uses EN 50155, the power industry IEC 61850, software for medical devices should follow IEC 62304. The electronics industry has opted for a common reporting format in a protocol called GEM-SECS.

Programs that have been written for Programmable Logic Controllers have long followed the IEC61131-3 standard. The purpose of the standard is to achieve source code portability between controllers from different manufacturers, but it also specifies how the parsing of the source code should be done, so that the resulting compiled code is executed in the same order in different controllers. However, this does not cover all aspects of control like synchronization and multi core type control applications. New standards are emerging, such as IEC 61499, etc.

Finally, it should be noted that in industrial automation many of the simplest problems are considered solved. This is not saying that there is no room for improvement. Current development centers around the merging of open source technologies in industrial applications, especially those that have been developed for cloud based connectivity. However, it can be argued that such development is just an improvement of current technologies, not principally new. Some researchers argue, that the rate of innovative new development has tapered off because new ideas tend to be more expensive to implement. A paper by Bloom et al. (2017) in a cooperation between MIT and Stanford discusses this.

8. Modelling

As a preamble to the modelling the author would like to point out that processes traditionally have been divided into discrete and continuous (Chapter 2.1). Discrete processes like assembly type manufacturing has been handled as one type and continuous processes like refinery crackers have been described as continuous. This thesis argues that this division is unnecessary from a complexity modelling standpoint. The reason is that all processes studied here can be described as having boundaries that can be approximated to being linear. Most control system models will assume that the process is within the control authority, and it is rare to have the behavior outside the control authority as part of the model description. The control model just ceases to be applicable outside the boundary conditions. Since many control systems continue to be in operation even in those situations, there is a case to study them for these situations. For example, most processes that are categorized as continuous are in fact discontinuous. The traditional continuous processes are just an approximation since they represent processes that stay in the same set of states for a long time, but not forever. The floatglass process by Pilkington in Halmstad (Sweden) was in uninterrupted operation for 14 years but that is probably a record. The complexity modelling presented in this thesis does not make a distinction, but can be used for analyzing all types of automation systems.

The author argues that there is good cause to analyze the boundaries of control ***before*** a control system is selected/designed.

Many systems that we are dependent on, such as transportation, power or pipe grids, are in fact in continuous operation. However, they are dynamic from a modelling state standpoint, as the combined resources fluctuate in a non-static pattern and can therefore be considered as continuous processes with ***n*** states.

The modelling of data poses a particular problem because most databases cannot operate continuously indefinitely. They need time to reorganize and create backups if not data access times are to expand with the size of the database. In other areas, service can be interrupted but this is not advisable for a road network. Especially, it is complicated to change the database structure while it is in operation. This is called a *schema* for a relational database. Blockchain type of databases may help, but no blockchain database has yet been designed to handle an indefinite number of transactions pertinent to a single unit described by its data. A blockchain is a linked list of data records (blocks), that are linked using cryptography.

8.1 State models

When modelling a process, a common method is to define which states a process can be in and then determine the conditions that causes the process to change state. This area has been extensively studied in the IT domain, in particular with regards to programming. For industrial applications, there are two main areas that govern how production processes are controlled through computers.

The most common model by the number of applications is discrete, where a transition is dependent on an ***event***, such as a timer elapsing, a number of conditions met or some other numerable condition. The important factor here is that the decision to proceed is binary, a go/no go decision.

The other application is the more traditional model, developed first for process industries, where the precise timing of the event is not known, but dependent on an analog condition, such as the filling of a tank to a certain level. In principle the system could wait a long time to act if a particular condition is never met.

There are good descriptive systems to describe individual subsystems, such as Modelica. Models have been studied by Li and Wieringa (2004). However, many systems in different application areas share common traits, and therefore there is a lot of interest in meta modelling. This has been studied by large automation companies because it offers an opportunity to create control programs automatically. One such example is xUML.

8.2 Object-oriented models

The classical way to model a production system data structure is by displaying it as a triangle, shown in Figure 8.1. At the base we will find sensors and signal conditioning. The next layer is comprised out of control systems such as PLC´s (Programmable Logic Controllers) and microcontrollers, which then are supervised by HMI (Human Machine Interface) computers and operator stations. The actions from these are coordinated by other computers of type MES (Manufacturing Execution Systems) and on the triangle top section we will find ERP (Enterprise Resource Planning) and MRP (Material Resource Planning) systems.

The model must accommodate the fact that modern production methods generate vast amounts of data. A small factory generates more transactions than the largest bank or insurance company. A single integrated unit like a large diesel locomotive can generate terabytes of data, within a quite limited time frame. A recent survey of generated data volumes is provided by Oussos et al. (2018).

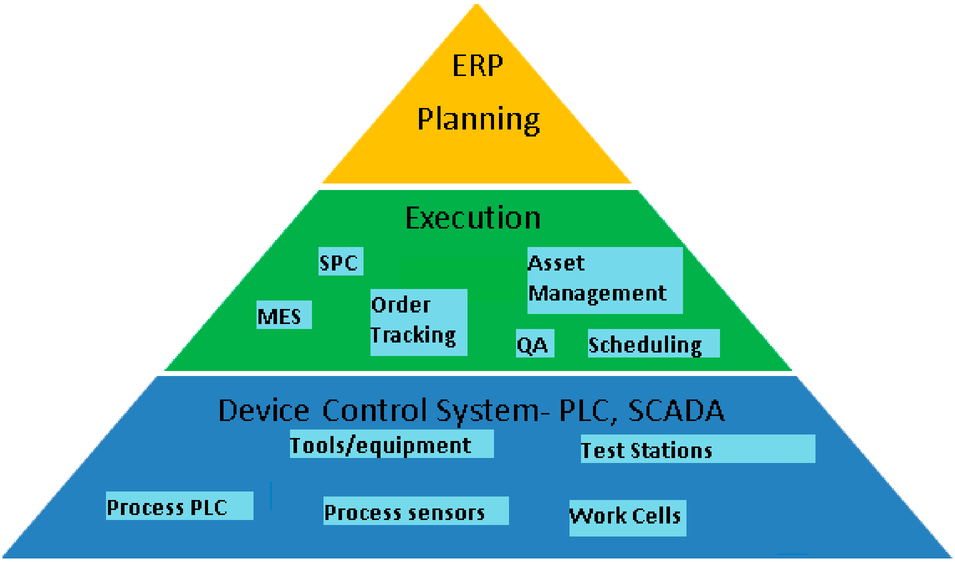

Figure 8.1 Traditional representation of a production system. Picture from Automation Primer.

Traditionally, large aggregation of data was not a problem, because as data flows upwards in the triangle through the layers above, then the number of transactions is reduced and the size of the data volume is reduced by methods like filtering, averaging and aggregation. Thus, the number of transactions becomes fewer while each transaction contains more information which in turn will be valid for a longer time period.

However, the introduction of new data models, such as publish/subscribe web services, means that the situation has now changed. The triangle has become a pyramid, where each data generator becomes a data server, which publishes data orthogonally in the triangle. Now every data server unit can talk to any subscriber at any level. This has been studied by Wagner (2002).

8.3 Cloud-based systems

The emergence of access to centralized computers (servers) via wireless networks has made it possible to design new types of system models. The general nature of these systems is that they belong to the integrative complexity category. Their complexity increases by the number of participating local systems, but the cloud design structure makes the model dynamic.

The most common design is one where a logical system has the capacity to cooperate or interact with the server application. The output from the local system is provided through publishing the availability, and the server application subscribes to the data.

One example of such a system is the power grid operator Ellevio/Fortum, which has a scheme where an owner of a private home is invited to place photovoltaic cells on the roof. Fortum will subscribe to the kWh´s generated during the day and return them to the house owner when he requires. The power grid in this case acts as the battery for the house owner, who does not have to invest in energy storage facilities.

Another case is the cloud-based SCADA application in the town of Tampere (Finland), where a lake level is controlled via a precipitation model fed by data from geographically distributed level measurements.

The data exchange of web connected systems has been extensively studied at many universities, for example Boualem et al. (2004) and at research institutions such as TEKES by Delamer et al. (2007).

The area of cloud based industrial system models is rapidly evolving, and it is easy to envisage new areas of complexity. It is not possible to put forward the concept of integrative complexity without mentioning systems that are connected via web exchange of data. Already there are systems that are controlling certain functions via setpoints generated from the Internet.

One such area of concern is that in many systems the boundary conditions are not fixed. If, for example, a number of independent power providers, such as wind generators, simultaneously decide to provide or withdraw capacity from the grid the resulting effect could exceed the designed voltage limits allowed. Many power generation systems in the Nordic countries are connected to the kilowatt-hour pricing structure of Nordpool, and it is easy to envisage the cumulative effect of hundreds or thousands of interconnected power systems using almost identical open source control software. One such boundary is the price, which can become negative resulting from the interconnectivity of producers.

In the example of the power grid the local systems are physically connected, but systems in a cloud application may display integrative complexity even without being physically coupled.

Many products can be manufactured in small "factories", such as 3D printers. It is easy to envisage a number of cloud connected production facilities that, together form a "mega-factory", where production can be changed rapidly by downloading production recipes to different participating machines.

Already this is to some extent already practiced by large corporations. The truck manufacturer Scania can produce identical trucks in several factories, and so can some manufacturers of micro ovens.

However, the advent of independent collaborating small manufacturers is creating new types of production models. One example is the apparel company Zara, using a network of contract seamsters to produce clothes at the same rate as the sales. The sewing machines are hand operated for most products, but production of articles like socks are generally fully automated.

8.4 Real life modelling examples

Very few processes have been modelled in extenso. The most common situation is where modelling has been used for simulation, so that potentially dangerous process conditions could be avoided or where practical testing is difficult or expensive. Examples are nuclear applications, development of aircrafts or submarines. Modelling is also used for automated stock trading, which is not an industrial application but automation control methods are now proliferating into many new application domains.

An example is the modelling of biological systems. In the definition of this thesis, this is a case of functional complexity. One example is the main endocrinical subsystems involved in diabetes. A model was made of the blood glucose level, which is affected by the insulin production in the pancreas, physical activity, body consumption, renal and liver functions as well as the sugar/carbohydrate content of food intake. This model can predict the glucose level with good accuracy and has been reported by Cescon and Ståhl (2009).

An example of operational complexity is the modelling work DYMASOS at the INEOS plant in Cologne, Germany (www.dymasos.eu). This is a huge site, covering several square kilometers. The production site is comprised out of more than 16 interconnected plants with thousands of sensors and thousands of actuators. Several hundred tons of raw materials are processed per hour, and the plants use several hundred tons of steam per hour. The interconnected networks deliver electricity, fuel gas, steam (at different pressures), nitrogen and intermediates. The INEOS model is comprised out of thousands of differential algebraic equations.

One recent example of a product based on an integrative complexity model is ABB´s OPTIMAX® .The product model used is highly similar to the complexity model proposed in this thesis. The system provides advanced model based control products for power generation and water utilities. Plant models are typically formulated in Modelica and deployed through FMI 2.0 (Functional Mock-up Interface). The optimizing control applications maximize the efficiency and provide more flexibility to large conventional power plants that face frequent load ramps and start-ups. Moreover, they aggregate small renewable units to large virtual power plants. This enables renewables to provide grid services like power/frequency control, achieving grid stability despite of high penetration of renewable power and raising revenues. ABB uses several compatible Modelica tools, including OpenModelica, depending on specific application needs. The system is currently installed to monitor the power grid of an island outside South Africa.

9. Conclusions

The first conclusion is that most complex situations cannot be validated in real life situations. This is because the system owners will not allow testing of real boundary conditions. If the boundary condition is sticky, then the test may result in a system failure. An extreme example was the analysis of risk at the large hadron collider at CERN, in order to avoid accidentally creating a black hole.

The second conclusion is that the analysis of control system limitations can be used *in a wider context*, e.g. applied to any system that has control signals with the purpose of maintaining control within certain setpoints or used to match demand to capacity. Many systems have a finite capacity, for example passing vehicles per hour, and various control methods are implemented to maximize use of the resources while minimizing risk for bumping into the limit (the boundary condition). One example is traffic on major highways, capacity increases until it is suddenly cut due to a disturbance caused by a small event.

A third conclusion is that the approach can be used to analyze tradeoffs between outputs. A speed bump in a road will reduce vehicle velocity, but will increase pollution and vehicle wear. An artificial road contraction will decrease speed but increase the risk of small accidents.

The fourth conclusion is that the qualitative approach proposed appear to be working in practical tests. So far, functional, operational, integrative and recursive complexity corresponds to properties inherent in all studied processes. The approach has a valuable effect in that the efficiency in the design process of new control systems is greatly improved, e.g. done faster.

Probably the most important finding may be that sensors with standard reliability cannot be used when a system is over and above a certain complexity threshold. The system reliability is dependent on system size and complexity analysis can determine how big a system can be allowed to become before its performance degrades.

10. Suggestions for future work

The approach proposed in this thesis needs further validation. The easiest way to validate the conclusions is to apply the boundary modelling method on new applications.

There are yet unautomated processes that displays functional complexity. One such area is the electrodeposition of precious metals on to electrodes used in medical devices. The automation of this process opens the prospect of big cost reductions in the manufacturing of continuous glucose measurement (GCM) sensors.

With regard to operational complexity, there is a lot of possibilities to improve the system performance of energy systems with many interconnected sensors. One such application resulted in a patent application in cooperation with LU Innovation in May 2019, based on work done for this thesis. In another operational complexity application. Ongoing tests (2019) are done by the author at The Royal Institute of Technology (KTH) in Stockholm, testing energy flow to municipal heat exchangers, supported by Vinnova together with SUST in Stockholm (SUST = Sustainable Innovation, an organization started by Energimyndigheten in Sweden).

When systems grow, the analysis shows that system size has impact on the control of individual sensors. The sensors need to have higher reliability in large systems than in small. There are also users that are interested in high reliability. Currently tests are ongoing at the Swedish National Railway operator (SJ) on the X40 long train system using sensor design based on ideas from this thesis.

Integrative complexity offers a good opportunity in the IoT (Internet-of-Things) area, since many such applications span over many nodes. There are indications that iterative complexity merits a more detailed analysis. By connecting many systems via the Internet, society is now capable of creating self-amplifying loops. One such example, could be when systems with many interconnected nodes, for example. solar power generators. If they all react on the same setpoint, such as a publicly available purchase price per kWh, the boundary grid conditions could be reached. Automation is spreading to new areas in society. One such example is automated trading of different assets. The varying boundary condition is set by the availability of automated buyers. If such a boundary condition is reached, then the automatons programming can make the price of an asset fall precipitously in seconds without any reference to the value of the underlying asset, as defined in a balance sheet.

With regards to recursive complexity, recycling and aggregation of chemically stable micropollutants in closed loop water systems are obvious examples, and studied for contained systems, such as space stations and small isolated ecosystems. However, besides the obvious systems there are other systems that are more hidden. The principle is the same as when one places a microphone close to a loudspeaker, which has been connected to the same microphone through an amplifier. Seemingly inconspicuous actions, such as running a production optimization algorithm against a measured demand can cause the production capacity to be reduced if the process has time lags in its model which are longer than the optimization frequency. Thus, recursive complexity can affect process boundary conditions such as capacity. These mechanisms are not sufficiently investigated.

Besides the practical applications described above, there could also be other areas that are more academic but still interesting. Determining the area of control created from a particular transfer control matrix is one such area. It should be possible to auto-determine the multidimensional polygon using a Turing box software technique (https://turingbox.mit.edu).

References

Abbate A., Katoen J-P., Lygeros J., Prandini M. (2010). Approximate model checking of stochastic hybrid systems. *European Journal of Control* 16(6), 624-641.

Anderson D.J. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press, Sequim, Washington, USA.

Barbucha R., Kocic M., Mizeaczyk G., Koziol G., Borecki J. (2008). Laser Direct Imaging of tracks on PCB covered with laser photoresist. Centre for Plasma and Laser Engineering, The Szewalski Institute of Fluid Flow Machinery, Polish Academy of Sciences, Gdansk, Poland.

Bloom N., Jones C., Webb M., Van Reemen J. (2017). Are Ideas Getting Harder to Find ?. NBER Working Paper No. 23782, Cambridge, MA., USA.

Boualem B., Fabio C., Halvard S., Farouk T. (2004). Abstracting and enforcing web service protocols. School of Computer Engineering and Engineering, University of New South Wales, Sydney, Australia.

Brunnemeier M.K., James H., Landau J-P. (2016). The Euro and the battle of ideas. Princeton University Press, Woodstock, Oxfordshire, UK.

Cescon M., Ståhl F., Landin-Olsson M., Johansson R. (2009). Subspace based Model Identification of Diabetic Blood Glucose Dynamics. Department of Automatic Control, LTH, Lund University, Sweden.

Delamer I., Martinez-Lastra J.L. (2007). Loosely-coupled Automation Systems using Device-level SOA. Institute of Production Engineering, Tampere, Finland. IMPRONTA Research project transaction by TEKES, Finland. 5th IEEE International Conference on Industrial Informatics, Vol. 2.

Fröhlich P., Hu Z., Schoelze M. (2002). Imposing modeling rules on industrial applications through meta modeling. ABB Forschungszentrum, Ladenburg, Germany, ER workshops 2001, Springer Verlag, Heidelberg, Germany.

Fässberg T., Fasth Å., Hellman F., Davidson A., Stahre J. (2002). Interaction between complexity, quality and cognitive automation. Chalmers Technical University, Gothenburg, Sweden.

Gillblad T., Olsson G. (1977). Computer control of a medium scale wastewater treatment plant. *Prog. Wat. Tech.* 9(5), 427-433.

Gillblad T., Olsson G. (1978). Implementation problems for activated sludge controllers. Symposium on Instrumentation, Control and Automation in Wastewater Treatment, International Association for Water Pollution Research and Control, London, UK.

Gillblad T. (1985). Use of local area networks in Tank Farm Automation. Worldflow ´85, London, UK. Presentation done in conjunction with work for the International Organization for Standardization (ISO).

Gillblad T. (1987). Geographically distributed Control Systems. Presented in Oslo 1987 by Texas Instruments at the Oil Industry Trade show (ONS).

Gillblad T. (1988). Hydrostatic tank gauging. Presented by Texas Instruments at Het Instrument in Utrecht. Presentation done in conjunction with work for the International Organization for Standardization (ISO).

Gillblad T. (1989). Jaugeage Hydrostatique. Presented by Texas Instruments at the Conference des Mesures a Nice. Presentation done in conjunction with work for the International Organization for Standardization (ISO).

Gillblad T., Johansson J. (2016). Efficient on-line wireless optimization of diesel deliveries via cloud service. ENER2016 conference, London, UK.

Hamburg A., Kalamees T. (2017). Method to divide heating energy in energy efficient building without direct measurement. 11th Nordic Symposium on Building Physics, NSB2017, Trondheim, Norway. *Energy Procedia,* pp 45-50.

Hazlewinkel M. (2001). Normal Distribution. Encyclopedia of Mathematics, Springer Nature, Basel, Switzerland.

Henze M., Harremoës P., la Cour Jansen, J., Arvin E. (2002). Wastewater Treatment. Biological and Chemical Processes. Springer Verlag, Berlin and Heidelberg, Germany.

Henze M., van Loosdrecht M., Ekama G., Brdjanovic D. (Editors) (2008). Biological Wastewater Treatment – Principles, Modeling and Design. UNESCO, IWA Publishing, London, UK.

Holland H. (1997). From Chaos to Order. Helix Books, Addison-Wesley, Redwood City, CA., USA.

Hyötieniemi H. (2003). Emergence and complex systems. Towards new practices for Industrial Automation. 4th Conf. on Intelligent Processing and Manufacturing of Materials, Helsinki University of Technology, Finland.

Johansson J. (2007). Risk and Vulnerability Analysis of Large Scale Technical Infrastructures: Electrical Distribution Systems. Licentiate Thesis, Dept. of Industrial Electrical Engineering and Automation, Lund University, Sweden.

Jones D.S.J., Pujado P.R. (2006). Handbook of Petroleum Processing. Springer Science and Business Media, Zürich, Switzerland.

Johnson N.F. (2009). Chapter 1: Two´s Company, three is complexity. Simply complexity: A clear guide to complexity theory. One world Publications, Oxford, UK.

Li B., Sansavini G. (2017). Energy market impact on the risk of cascading outages in power systems. Reliability and Risk Laboratory, Zürich, Switzerland.

Liptak B.G. (2002). Instrument Engineer´s Handbook. CRC Press, Boca Raton, FL., USA.

Li K., Wieringa P. (2004). Interactive modelling software for supervisory control of industrial processes with increased complexity. IEEE 2004 Conference on Systems, Man and Cybernetics.Omnipress, WI.,USA.

Lenz M., Rhodin J. (2011). Reliability calculations for complex systems. Masters Thesis, Linköping University, LiTH-ISY-EX-11(4441-SE), Sweden.

Ljungblad S., Franzèn T., Holmsten M. (2015). Temperaturmätning i vätskeflöden. Rapport 118, Svensk Fjärrvärme AB.

Krus P. (2018). Modelling and Information Entropy of Design Spaces. Open Modelica Workshop at Linköping University, Sweden.

Kuo W., Zuo M.J. (2002). Optimal Reliability Modeling, Principles and Applications (1st ed.). John Wiley & Sons, New York, NY, USA.

Olsson G., Rosen C. (2005). Industrial Automation - Applications, Structures and Systems (2nd ed.), Lund University, Lund, Sweden.

Olsson G. (2015) Water and Energy –Threats and Opportunities (2nd ed). IWA Publishing, London, UK.

Oussos A., Benjelloun F-Z., Ait Lahcen A., Belfkih S. (2018). Big data technologies: A survey. *Journal of King Saud University* 30/4, 431-448.

Pasche M., Persson M. (2008). Impacts of product complexity on organizational complexity. A case study at Volvo Car Corporation from a manufacturing perspective. Chalmers University of Technology, Sweden. 15th International Product Development Management Conference, Hamburg, Germany, June 29 - July 1, 2008.

Poincaré H. (1902). La Science et l´Hypothèse. Flammarion, Paris, France.

Poisson S.D. (1837). Probabilité des jugements en matière criminelle et en matière civile, précédées des règles générales du calcul des probabilités, pp137-143. Bachelier, Paris, France.

Popper K. (1959). The logic of scientific discovery. Basic Books. Online at https://www.questia.com/library/1301513/the-logic-of-scientific-discovery

Radestock M., Eisenbach S. (1996). Coordination in Evolving Systems. Dept. of Computing, Imperial College of Science, Technology and Medicine, London, UK.

Scholten B. (2007). A guide to applying ISA-95 standard in manufacturing. ISA Society Publishing, Research Triangle Park, Durham, NC, USA.

Sha L. (2001). Using Simplicity to Control Complexity. *IEEE Software Journal* 18/4 Jul./Aug. 2001.

Svensson J. (2006). Active Distributed Power Systems. Functional Structures for Real Time Operation of Sustainable Energy Systems. PhD thesis, Dept. of Industrial Electrical Engineering and Automation, Lund University, Sweden.

Wagner T. (2002). An agent-oriented approach to Industrial Automation Systems. Institute of Industrial Automation and Software Engineering. Proceedings of 3rd international Symposium on Multi-Agent Systems, MALCEB 2002, University of Stuttgart, Germany.

West G. (2017). Scale: The universal laws of growth, innovation, sustainability and the pace of life in organisms, cities, economics and companies. Penguin Press, New York, NY, USA.

Wilkins J. (2018). Improve supply chain management with AI technologies. *Control Engineering Magazine,* Aug. 29, 2018.

Yanovski N.S. (2016). The outer limits of reason, what science, mathematics and logic cannot tell us (5th ed.). MIT Press, Cambridge, MA., USA.

Application references

The thesis has been based on the author´s notes from a large number of industrial applications, some of which are listed below:

Example no	Source
2.1	BP refinery at Dunkerque, France
3.1	Habia cable factory in Söderfors, Sweden
3.1	XSYS paint plant in Trelleborg, Sweden
3.3	Ericsson office building in Kista, Sweden
4.1	SSAB steel plant in Luleå, Sweden
4.3	The wastewater treatment plant at Duvbacken, Gävle, Sweden
4.4	Varta battery factory in Hultsfred, Sweden
5.3	Esso Fawley refinery in Southampton, UK
5.3	Nynäs Petroleum lubricants refinery in Nynäshamn, Sweden
5.5	Intel semiconductor plant in Chandler, USA
5.6	The V-TAB printing press in Norrahammar, Sweden
5.7	LKAB iron ore mine in Malmberget, Sweden
5.8 + 5.9	Arcelor Mittal steel mill in Avilés, Spain
5.10 + 5.11	Älvsbyhus house factory in Älvsbyn and Vålberg, Sweden The Ringsted House factory in Ringsted, Denmark The A-hus house factory in Kungsbacka, Sweden
5.12	Sibes foundry in Gnosjö, Sweden
5.14	Scania truck cab assembly in Oskarshamn, Sweden
5.15	Automation direct warehouse in Alpharetta, USA
5.16	Arctic paper paper mill in Munkedal, Sweden
5.19	Tetra Pak Research & Development, Lund, Sweden
5.19	Celloplast factory in Norrköping, Sweden

Appendix

This appendix contains an example of complexity analysis according to the principles put forward in Section 7.2.

The appendix is divided into two parts, firstly the transfer function is constructed, then the complexity is analyzed using a measurement method with inherent complexity.

The EU *Directive 2012/27/EU on Energy Efficiency,* Articles 9-11, with respect to thermal energy supplied from collective systems stipulates that all new apartments shall be equipped with individual energy consumption measurement devices.

This directive has been only slowly adopted since the instrumentation cost is significant when compared with the value of expected energy savings. The Swedish building industry has been granted a delay in implementing the directive. However, it has been shown that information about consumption impacts the behaviour of people so that on average, a 30% reduction is achieved by metering. Therefore, eventually, it is expected that metering will be introduced on a larger scale in Europe.

Currently invoicing of energy consumption is most often calculated based on the areas of the respective apartments in a building.

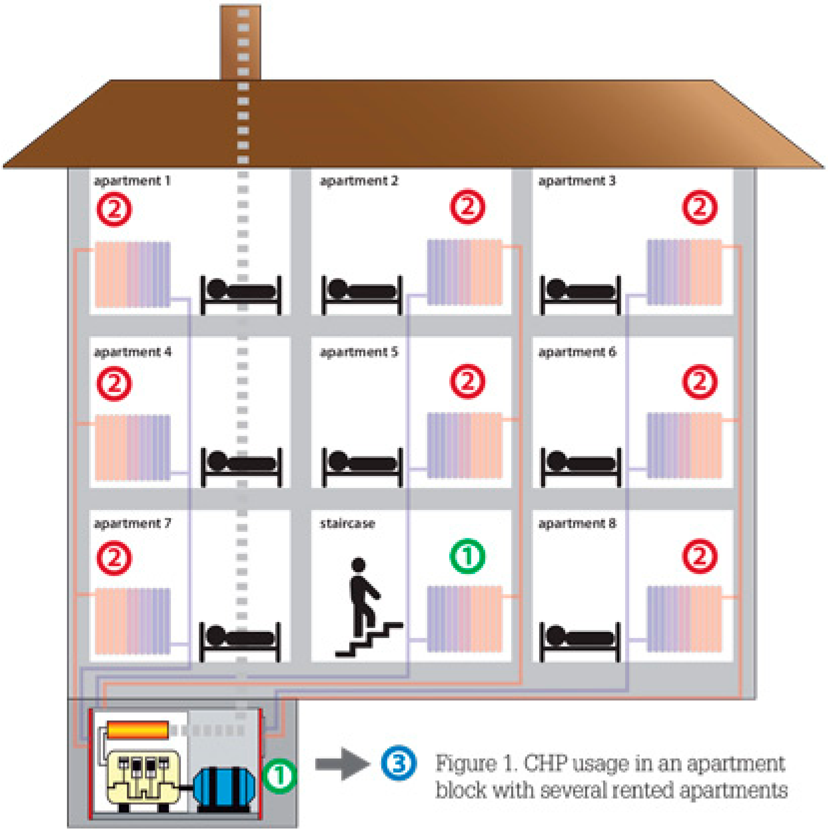

Figure A 1. Cut through view of an apartment building.

An apartment building, with *n* apartments, is sketched in Figure A 1. Energy (heat) is provided from the municipal grid through two heat exchangers.

Assume that there are *n* apartments, with individual apartment areas A_k, where k=1,n

A = Total apartment area, where $A=\sum_{k=1}^{n} A_k$

B = Area of common surfaces

A+B= total building area

The most common way of splitting the energy cost between the tenants is made by apartment area, according to the formula

I_k= E_C (A_k/A + B/n)

where invoiced energy I_k is based on apartment area, E_C= cost per kWh of energy.

This type of calculation is common, but it has the drawbacks of giving not very fair invoicing between tenants and it does not give any incentives to conserve energy. Also, if the measurement of supplier energy from the municipality is different from what is locally measured through individual meters, then there may be a difference to split. Therefore, the most common method is to use the invoiced amount W_{EP}

so that W_{EP}== $E_{INV} \sum_{k=1}^{n} I_k$

where the invoiced cost E_{INV} replaces E_C. Many municipal heat suppliers index their cost to the price of fossil fuels, which discourages investments in technology like heat pumps. Also, there is little incentive for an individual flat occupant to install energy savings like triple glazing or conserve energy by other methods.

In order to improve fairness, Hamburg and Kalamees (2017) have introduced a model which takes more factors into account. Their formula for the total energy balance of an apartment is

W_{EP}=W_{HEAT}+W_{VENT}+W_{DHW}+W_L+W_{EQ}+W_{APP}

W_{EP} stands for total energy consumption, W_{HEAT} for energy for space heating, W_{VENT} for energy for heating of ventilation, W_{DHW} for heating hot water for consumption, W_L for electric energy for lighting, W_{EQ} for electric energy for household appliances, W_{APP} for energy for building service systems.

This model will create a higher level of fairness, but the number of meters required represents a costly investment and some of the factors are an order of magnitude smaller than the others. Also, there is no financial incentive to conserve energy.

This is the reason why the EU has made individual metering mandatory in new buildings, as it looks at the real consumption and as this gives the tenants a financial incentive to conserve energy.

When metering is introduced, then most commonly a flow meter based on for example the Doppler effect is installed, and a volumetric meter is used to measure the amount of water used for consumption. The calculation becomes

f_w = fresh water flow (for consumption)

f_{Ew} = water used for heating

T_{in}= temperature of incoming water to the building

T_{out}= temperature of outgoing water from the building

The total amount of energy provided during 24 hours is:

$$e_k=\int_{t=0}^{24} f_w(t_{in} - t_{out})\ dt+\int_{t=0}^{24} f_{Ew}(t_{in} - t_{out})dt$$

If the price for energy is c_k per kWh and the price of consumed water is ew_p/m^3 then the invoice per day becomes

$$I_k = e_k \ . \ c_k + ew_p \int_{t=0}^{24} f_{2k}\, dt$$

The price could vary between different apartments depending on consumption, hence the *k* index factor.

The total energy invoiced is

$$\mathbb{E}=\sum_{k=1}^{n} I_k$$

Total invoice will include consumed water. If the water price is ew_p/m^3 and cost for hot water used will be

$$I_{wk}=w_p \int_{t=0}^{24} f_{2k}\, dt$$

And of course

$w_p = \int_{t=0}^{24} f_w \mathrm{dt} = \sum_{k=1}^{n} I_{wk}$

If a new model of individual energy measurement is introduced, according to the EU directive then the model becomes more complex. This method use two flow meters based on the Venturi principle, as shown in the Figure A 2 below.

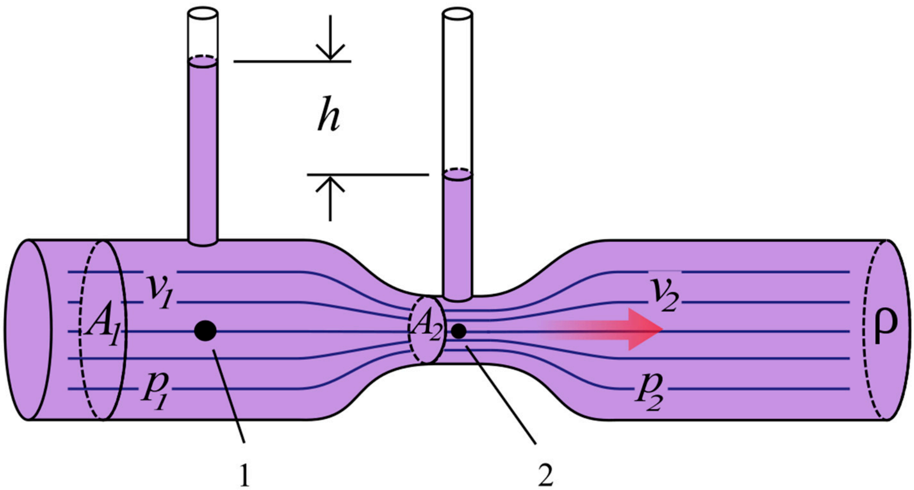

Figure A 2. Venturi flow meter principle. Picture from Wikipedia.

Each flow meter has two pressure transmitters and two temperature sensors. Each apartment has two flow meters and a sensor measuring the temperature of the outgoing water that has been used for heating.

The temperatures for water used for heating in apartment *k* are designated, in the direction of flow

t_{1k}=first temperature measurement

t_{2k}=second temperature measurement

The temperatures for consumed hot water in apartment *k* for heating are designated, in the direction of flow

t_{3k}=first temperature measurement

t_{4k}=second temperature measurement

The outgoing temperature of water used for heating is designated t_{5k}

Total energy consumption in one apartment during 24 hours is:

$$e_k = \int_{t=0}^{24} (f_{1k} \cdot (\frac{t_{1k}+t_{2k}}{2} - t_{5k}) + f_{2k} \quad \frac{(t_{3k}+t_{4k})}{2}) \mathrm{dt}$$

t= time in hours

The parameters above can be put in a single transfer function matrix, with the simplified layout

$$\begin{pmatrix} I_k \\ I_{wk} \\ I_c \end{pmatrix} = \begin{pmatrix} f_{1k} & f_{2k} & f_E \\ w_{1k} & w_{2k} & 0 \\ 0 & 0 & c_k \end{pmatrix} \begin{pmatrix} t_{1k} \\ t_{2k} \\ t_E \end{pmatrix}$$

Complexity analysis

Going from simplified invoicing based on apartment area to individual measurements in each apartment introduces complexity.

The functional complexity emanates from the fact that the boundary conditions for the temperature measurements are narrow. Most flow measurements are also nonlinear. The operational complexity comes from the fact that energy consumption varies with ambient temperature and the habits of the apartment occupants. Finally, the system displays integrative complexity in that a large building has many interconnected sensors, who together contribute to the total consumed energy.

In order to analyze this complexity, it is necessary to study the reliability of the sensors. Such a system is considered ***serial***, from a reliability calculation standpoint.

Reliability formulas are to be found in the thesis by Lenz and Rhodin (2011) and the one that describes serial reliability is

ϕ (x) = min (x_1 ……x_k) = Πx_k (k=1….5k)

The formula calculates the combined reliability for a system where there are k apartments and 5 sensors per apartment (two flow and three temperature). Each sensor reliability is designated x, and is expressed as a value between 0 and 1. The respective reliabilities are multiplied in a serial system.

Reliability is normally constructed as a Boolean value where 0 = failure and 1 = correct operation. However, this is not the case which we are concerned with here. We are interested in the risk that a sensor is outside its normal operating boundary condition, which we will call MTBOB (mean time between out-of-boundary). Temperature sensors are normally very reliable in that their MTBF (mean time between failure) is high.

Both temperature and flow sensors risk measuring incorrect values, but in most cases of energy measurement, it is temperature errors that have the bigger impact on invoicing. This is because of Stefan´s law, stipulating that black body radiation is proportional to the Kelvin temperature to the power of 4. A higher incoming temperature of 3 degrees will result in increased radiation of energy from convection radiators by approximately 4%.

For a Pt100 or Pt1000 temperature sensors, the annual drift in accuracy is in the order of 0.05 °C per year. Pt100 and Pt1000 stand for resistance thermometry, using platinum resistors of 100 or 1000 ohm respectively. The resistance changes linearly with the temperature in Kelvin, and is measured via a comparator circuit (a Wheatstone bridge). However, there are many other factors that can influence the measurements, such as differences in sensor characteristics originating from the manufacturing of the sensors, effects of corrosion, differences in linearity and also variations in the supply voltage to the measurement circuits. Also, when the sensor is placed in a sheath in the water pipe there is delay until a temperature change is registered. Digital measurements are limited due to the accuracy in the A/D converter and impurities in the water can result in deposits on the sensor surfaces. An overview of the most common temperature measurements can be found in Ljungblad et al. (2015).

The total effects of the abovementioned inaccuracy sources is obviously hard to gauge, but can be calculated indirectly.

For temperatures, the exact temperature T can be set to be the measured temperature T´ plus the estimated maximum temperature error.

$T = T' + O(\Delta t)$

The letter O stands for Ordo and is a measurement of how heavy the parenthesized parameter weights in the calculation. When the sum of the individual energy consumption measurements in all apartments in a building differs from the supplied amount with more than the maximum the system is designed for, then we know that complexity has created an out-of-boundary condition.

Example

> If a building has 50 apartments and 5 sensors per apartment, we can assume the individual MTBOB (Mean Time Between Out of Boundary) is 8 000 hours (approximately one year), then it will take approximately 32 hours until the invoicing system is out of boundary.
>
> If the actual calculation shows another value, for example 60 hours, we know that the mean time until a single sensor has reached a boundary condition is approximately 2 years.

Because integrative complexity is insidious in nature, this example shows that most systems we rely on are probably not within their design specifications, but since the maintenance and calibration work required to keep these systems in good working order is so onerous, we put up with poorly working large systems.

A better model would be to introduce systems that are more resilient to effects of complexity. One such would be to introduce sensors with in-build redundancy. Such systems are considered having standby systems properties. The reliability calculations differentiate between hot standby, warm standby and cold standby systems, described by Kuo & Zuo (2002, pp 129).

A flowmeter based on the venture principle can have multiple temperature measurement points.

Reliability calculations of standby systems differ radically from systems with single points of failure, but results in most cases in several orders of magnitude increase in reliability.

If a flow rate is measured with pressure difference meters, then each meter contains two pressure sensors and two temperature sensors. As a consequence, there will be $9n$ sensors added for a building with n apartments. For an apartment building with 50 apartments, this means 200 more sensors than in a system without redundancy.

However, this is not the reliability in the common sense, because there are physical constraints, based on the logic that all temperature values must be within logical boundaries. The temperatures are designated:

T_{in}= temperature of incoming water to the building

T_{out}= temperature of outgoing water from the building

$$T_{out} < t_{xk} < T_{in}$$

where *x*=1,2,3 or 4

If a sensor value is higher than T_{in}, then the meter is showing that heat is generated during distribution, which is illogical. Some t_{xk} could theoretically be larger than T_{out}, but not with a large amount, since the flow speed means that used water will leave the building in seconds.

The technical MTBOB of a single sensor may be fairly high (in the order of 10^6 hours). However, the chance that it will drift, be out of calibration or sourced with a too low voltage in the measuring Wheatstone bridge is much higher. If it is "out of sync" with the temperature used for the energy supply invoicing, then the invoicing, ***regarded as a process***, is suffering from integrative complexity.

Now we introduce the boundary conditions, if a sensor t_{1k} is outside its boundary and t_{2k} is within, then the value of t_{1k} is replaced with t_{2k}. The opposite is also true.

The algorithm for this is

if $t_{xk} \notin [t_{out}, t_{in}]$, and

$t_{(x+1)k} \in [t_{out}, t_{in}]$, then $t_{xk} = t_{(x+1)k}$

Similarly, for hot water consumption, if a sensor t_{3k} is outside its boundary and t_{4k} is within, then the value of t_{3k} is replaced with t_{4k}. The inverse is also true.

The algorithm for this is the same as for water used for heating, e.g.

if $t_{xk} \not\in [t_{out}, t_{in}]$ and

$t_{(x+1)k} \in [t_{out}, t_{in}]$ then $t_{xk} = t_{(x+1)k}$

Reliability can be further enhanced if both t_{xk} and $t_{(x+1)k}$ have illogical values, or are malfunctioning they can be replaced with $t_{(x+2)k}$ and $t_{(x+3)k}$.
And, finally, all sensor values can be replaced with T_{in} if all temperature sensors are malfunctioning at the same time, in which case the invoicing process degrades into method based on standard temperatures as is the case of invoicing based on areas.

Example

> The risk that 4 sensors connected to the same flow should malfunction is much lower than for a single sensor and if software is implemented using the results from complexity analysis, then the system mean time between out of bounds, e.g. calculating invoiced energy incorrectly, rises from approximately 50 hours to an estimated 250 000 hours. It will no longer be determined by the boundary conditions, but be completely determined by the technical lifespan of the sensors.

This logic also applies to the measurement of flow. Since the energy is the flow multiplied with the temperature difference, then errors in the integrated measurement on total flow to a building can result in incorrect invoicing in for example consumed water. However, since the temperature difference relative to the absolute value of the temperature is much smaller than the absolute value of the flow vs. its range, it is errors in the temperature measurement that are most likely to cause errors in invoicing.