EARLY DETECTION OF ABNORMAL EMERGENT BEHAVIOUR

Lambert Spaanenburg
Dept. of Electrical and Information Technology
Lund University / LTH, Lund (Sweden)
Email: lspaanenburg@ieee.org

ABSTRACT
Emergent behaviour has become a plague of automation systems based on communication networks. Centralized monitoring of the network comes generally to late to suppress unwanted behaviour. It is required to mark the tendency towards state changes in a decentralized manner. The paper discusses the role of local awareness by inspection of the model learning behaviour of feed-forward networks. The correlated movement of weight changes over time provides a clear indication of such profound changes, as demonstrated by some initial experience in industrial automation.

1. INTRODUCTION
Emergent behaviour has become a major threat to modern automation. It was not so visible in the early days of automation, when a single machine was modelled and controlled. The model was developed to be robust enough for providing reliable results despite noisy, irreproducible and incomplete measurement data. Faults are modelled separately and Fault Diagnosis and Isolation (FDI) is based on classifying machine behaviour as normal or having a known and characterized fault [1], [2].

Classical abnormality detection starts from the single fault assumption. Faults are observed at the output, but can hardly be distinguished from other faults on the same path to the system output. When a fault is observed and leads to an operator alarm, the remedy is not always clear. The single fault may actually lead to a flood of alarms messages that annoys the operator, but does not always create disasters.

Fault observability should therefore not only address the ability to see a fault but also the amenability to rank its urgency and the agility to react accordingly. Without such measures, alarms tend to be simply ignored in practice and it is necessary to regularly perform a health check to bring the alarm flood back in proportion. Typical techniques as reviewed in [3] have led to the SilentControl system (Figure 1).

But the realm of automation has grown into networks. Abnormalities do not only give errors, but can also travel over the network to reach other machines. Most of these networks grow without an overall architectural vision but rather by means of a local preferential attachment. This lack of predetermined structure does not mean that there is no structure at all. On the contrary, it has been noted that the seemingly chaotic self-organisation leads to a clear structure with distinct properties, though different from designed networks [4], and display therefore emergent behaviour.

Figure 1: Excerpt from SilientControl monitor box [3].

The global structure of such networks with inherent communication delays has become notorious for abnormal behaviour. For instance, default distribution of a programming error as part of the maintenance procedure caused in 1992 the New-Jersey blackout. Only a couple of years later, the Allston-Keeler (July 1996) and the Galaxy-IV (May 1998) disasters gave rise to a concerted research activity on Self-Healing Networks [5]. A series of three disasters on the Electricity Grid in autumn 2003 (in respectively America, Sweden and Italy) suggests that little progress has been made. And these are only the tip of the iceberg [6].

Electronic networks (and especially wireless ones) are not exempt from emergent behaviour. Even when the design has been perfect, ageing & wear can develop in unknown ways. Moreover, embedded systems are becoming more of the reactive nature that makes abnormal behaviour likely to emerge. Where autonomous nodes work together, they tend to pass not only the good but also the bad news. Consequently special measures are required to make them ‘immune’ for sick neighbours [7]. Of course, the degree of immunity (or self-healing) must be dependent on the fatality of the sickness. The critical point is clearly how to differentiate between the need to globally adapt and the demand to locally block a fault effect from spreading.

In addition, redundancy allows overruling a malfunctioning part to decrease the fault sensitivity. It is usually eliminated to bring down initially product cost. However, an optimal design should exploit redundancy to the fullest, because it reduces maintenance costs over the lifetime of the
system [8]. Most effective is partially redundancy, though this will inevitably raise the need to contain emergent behaviour.

The paper proceeds as follows. In section 2 the concept of emergent behaviour is further elucidated. Ensuing section 3 discusses why a feed-forward neural network is well equipped to do this task. Lastly some experiences are discussed and conclusions are drawn.

2. EMERGENT BEHAVIOUR

Feature interaction over time between the network parts is the main cause of the problem. The lack of separation means that the state count of the network now equals the product of the state count of the individual machines, instead of the sum. Moreover, as the machines are individually designed, putting the machines into a network will make that a large number of the network states are new to the system. Though they are not likely to appear, their behaviour has not been inspected before: they seem to emerge out of the blue.

Embedded platform intelligence is required for distributed sensory networks that are composed of (im-) perfect components with (im-) perfect models. The number of possible states and state-interactions explodes with increased freedom in local throughput and local control & calibration, and a global model of the emergent behaviour of a system using local autonomous behaviour is not tractable. There is no consistent model for design-time validation.

Corrective models for set-point predictions of an industrial processing system show time-varying global disturbances (not observable by the mathematical models) that are present and have been suppressed. Without intent, emergent behaviour appears in such distributed systems due to nondeterministic computing and distributed indirectly controlled feedback algorithms. Consequently, a gap between the desired behaviour as physically understood and the actual behaviour comes to the surface. As illustration, the behaviour in Figure 2 is a consequence of the interaction between application and protocol layers in a computer network; hence it must be considered as emerging behaviour.

Figure 2: Residual error observations from model fitting the communication in a KPN telephony network.

In this paper we assume that the measurement data are already cleaned from false indications, as performed in [3]. Only actual faults exist and will eventually become observable. We see these as the result of parameter variations over dimensions not included in the model. Such extra dimensions cover a number of related models, of which the fault models are discrete samples. This brings the need for two essentially antagonistic views on the same reality (Figure 3): (a) the process model reflecting the proper operation of the network, and (b) the detection model reflecting the abnormalities.

Figure 3: The parameter space for novelty detection and isolation (NDI).

The former relies on the extraction of physical coefficients from model parameters [1]. The aim is to find the Degrees of Freedom (DoF) in the model that describe the process in full complexity while minimizing the least square error (risk minimization). Domain experts usually add as constraint that the model can be humanly interpreted. The relevance of a known disturbance can be built in by ensuring that the error on desired behaviour is a measure of the “faultiness” of the observed behaviour.

To handle unknown faults, the generic sensitivity (universality) needs to be improved by increasing the DoFs in the model. However increasing the DoFs conflicts with risk minimization. Simplifications may be called for to reach the desired separation of concerns. The justification for simplifications comes from properties on the independence of local processes allowing for modular models and linear behaviour. Indeed, if a system composed of multiple processes is in a stable equilibrium in its state space, linearization is allowed [9] and will break the dependence between sub-processes. Unfortunately these properties are insufficient as soon as the system drifts from the desired stable equilibrium (Figure 4).

Figure 4: Set-point becomes non-optimal when abnormality occurs.

The alternative proposed here is to break the dimensional dependence between model complexity and model risk to improve both overall sensitivity as well as confidence in
Figure 5: Some typical dependencies between parameter gradients in a non-linear redundant system.

Structure, resulting from chaotic sources or interaction of sources, cannot be identified from the data without an exact model; it is unobservable. A structural pattern, which emerges from the platform implementation, remains a problem. Structural dependencies may be hidden in the data and residuals locally, but they do appear as artefacts in sensory images, after data aggregation and cleaned through calibration and model fitting. Structural dependencies are not reduced through averaging; moreover cross-dependencies between effects in different and remote signal paths prevent full observability and will cause convergence problems in the solving process. Such problems are well known in learning of neural networks. Typically this occurs in dynamic equilibrium of a redundant neural network, as illustrated in gradient-gradient plots in Figure 5.

3. REFLECTIONS ON FEEDFORWARD ANN’S

The feed-forward artificial neural network (FNN) is a simple arrangement of basic neurons such that all signals will only move forward. Usually the FNN is layered such that the signals in one layer will only go to the next one and no layer moves forward. Usually the FNN is layered such that the arrangement of basic neurons such that all signals will only move forward. Such models need to match two opposing worlds by creating two different but dimensionally overlapping views on the same reality.

The difference between the actual and the target output of each neuron \( j \) in the last layer \( L \) is used to determine the error on the weights of the incoming signals, and they are slightly adjusted. This is repeated for all previous layers in order. The gradual adaptation of the weights is meant to converge to setting, wherein the network will have learnt the target behaviour.

The critical point is the assumption of convergence [11]. From a situation wherein the weights are initialized with small random number, convergence can always be guaranteed. When the weights are not small but pre-computed in the right order of magnitude, convergence will also not be a problem. However, when the weights are large and totally wrong, the learning behaviour is somewhat more questionable. This will notably show in modular neural networks.

Figure 6: Re-training a fully trained neural network shows a period of internal adaptation, followed by a chaotic attempt to fit the new data into the model.

When learning is not on the right track, there is every chance that the network will go through a random phase trying to find the correct direction. This is primarily caused by the fact that equation (3) assumes that the error propagation is continuous. When this is not the case, because a strange element has been inserted, the internal weight corrections will be so large that any pre-integrated knowledge will be removed. A typical example is shown in Figure 6.

When the system function is already continuous, the error propagation will gradually die down. Lets assume a 3-layer feed-forward network. Small adaptations of the weights between the middle (hidden) and last (output) layer can be made without major error propagation back. Usually the neurons in the hidden layer are said to contain the features, by which the output function is determined. This vector space has so many degrees of freedom that always a solution can be found by a slightly different vector composition from the same feature base.

Only when the feature base has to change, will the weights between the first (input) and the hidden layer change. The fact can be taken as the principle to detect early model changes. Though there is an appreciable amount of internal activity, the output does not necessarily change much. Hence, the fact that the neural network has to re-model for a changed reality does not become clearly apparent on the outside. Therefore the detection will be early.

We know now where to look, but still the question remains how the early detection can be made. The simple fact of detecting changes in the weight values is not enough, as some changes will always be there. We need to see small but still consistent change that moves hidden features!

Vector algebra gives a clue how this can be done. Adapting to a new model means that the required features have...
changed. Therefore the subsequent weight adaptations move the features gradually to the new desired location. This can be detected by looking at the $2^{nd}$-order derivatives of the weights in time. A consistent (read correlated) change of the weight movements will indicate a learning path being active, creating a better base for the capture of the current reality. An example is shown in Figure 7.

**Figure 7**: The distribution of gradient-correlations in a neural model for a Volterra-Lotka system in the equilibrium, i.e. converged to a noise level, horizontal axis is [0 1]. This shows strong dependencies between gradients; these are typical dynamics in the equilibrium.

### 4. DISCUSSION

Treatment of emergent behaviour requires the detection, diagnosis and accommodation of global disturbances. In model-based detection approaches, disturbances are diagnosed through parameter fitting of nominal process models. However emerging behaviour does not fit well in this scheme. The challenge is the early detection of emergent behaviour preceding diagnosis and accommodation.

Phenomenological approaches based on the process history [12] employ computational intelligence such as data mining for IT infrastructure [13]. Early detection requires a metric for profundness of emergent behaviour reflecting the impact feasibility of the desired functionality. In the data mining approach [13] the cost-function is derived from the agreed services level agreements. In the Lydia approach [14] the detection metrics are derived through propositional logic from the required functional behaviour. Lacking a parameterised behavioural model, such detection approaches do not provide the sensitivity needed for early detection.

For early detection we have proposed a model-driven black-box detection method based on dependencies between parameter gradients [15], of which the principles have been elucidated in section 3. This method has been developed over time in a number of practical cases, some we will shortly discuss here. Figure 2 comes from a feasibility study on the detection of attacks in a communication network. It is learned with a collection of normal traffic signals, and responds to anything out of the normal. It does not give any clue on the cause of the abnormality, but requires it to be consistent. This way it will not fire on the occasional novelty.

In a hot-strip mill, a long strip of heated metal is rolled through a series of mills (rather than just one) and thereby successively pressed to the desired thickness, simultaneously making it longer. In the process of pressing the steel plates the plates decrease in thickness from O(10cm) to O(1mm), and consequently the initial pace of a few m/s increases to O(100km/hr), and the force control need to respond rapidly to variations. The variations are measured directly after each of about 10 mills.

Hence each mill has its own control, set to a value that is derived from the global target by means of a physically plausible model for pressing metal shapes. The hot-strip mill operates on batches: operation is started, comes on steam, produces the lot, slows down and stops. The controller must handle all these different modes and the shady regions in between; or rather the global control task has to be divided into subtasks on individual mills in potentially different modes of operation.

**Figure 8**: Detection results of increasingly profound change: sensor disturbance (top), state perturbations (middle) and varying dynamical disturbances (bottom).

The operation of each individual mill is largely understood on physical principles but not in sufficient detail for a complete analytical model. Adaptations are necessary because all mills are slightly different in construction and react different to ageing. A regular stream of corrective models has been necessary over time to answer the increasing demands in production efficiency and environmental dependability.

It was therefore appropriate to try a different route. Figure 8 gives an overview of results for a correlation test for the prediction of a simple sine wave. The largest observed fraction of different correlations in the 25 cross-validation models over all connections barely tops the minimal observed fractions of different gradient-correlations for the saw-tooth and does not even come close to the minimum fraction of the different correlations for the shifted sine.

This and similar tests has given enough credibility to use the gradient correlation for inspection mill data. Off-line early abnormality detection has been successful to the degree that also quality changes during intermission of the operator attention were noticed. The link with the production line is however debatable, as emergent behavior is detected but not diagnosed.

### 5. CONCLUSIONS

We have discussed that emergent behaviour needs to be detected, before it spreads through the network and gets the network out of control. The classical central monitoring of black-box phenomena has clearly lacked effectiveness in the past. Regular maintenance helps to reduce the risks, but early
The essence. Engineering, but are limited to detection as reaction time is of the system from a non-functional part of the environment. Off-line guards are required to handle faults that creep into and are directly related to an existing software object. The layered focus over all control level.

Figure 9: V-chart for separated process and detection modelling, illustrating the dedicated but hierarchically layered focus over all control level.

The learning process creates a sense of history, but taking care of a non-linear dependence and without the need to reduce the number of observed parameters. This is of interest, where the process interacts with its environment. In our case there will be different structures in different views. Another difference is the fact that the neural network evades the need to perform an m-dimensional mapping and matching. Instead we have a simple mechanism that can be easily added to an existing software object as a guard mechanism.

We have to distinguish between two types of guards: off-line and in-line. The in-line guards are as discussed in [17] and are directly related to an existing software object. The off-line guards are required to handle faults that creep into the system from a non-functional part of the environment. This is depicted in Figure 9 as V-chart for the Power Grid control case [5]. This picture has a striking resemblance to model separation shown in Figure 3.

Most often, off-line network problems have a very basal cause. For instance, the quality of the electricity supply leaves nowadays much to be desired. Regularly, the supply is interrupted for a short time [18]. If this time remains below 10 ms, the human operator will not even notice it but the machines get affected at least to the degree of a noticeable reduction in lifetime. If the supply elapses longer, the electronic equipment gets deregulated. A much-advocated solution is the insertion of an UPS to bridge such gaps. This has caused a consequential problem in an Internet router, where the electronic equipment kept functional but the ventilator stopped, causing overheating of the system followed by a melt down. Future local supply systems can remedy this fault category, but still leaves other issues such as simultaneous switching untouched.

Apparently, network faults can become inserted at almost any time and place from other technologies than the mere electronic design. This makes multi-level modelling [19] a necessity to create a real robust operation.

REFERENCES