A Simulator for Log-structured Data in Flexible Assembly System Fault Prediction - DRAFT

Tero Keski-Valkama

Version: August 15, 2015

***Abstract*—**

***Index Terms*—**

I. Introduction

This paper describes a method for simulating flexible assembly system log-structured data sources including faults. Such simulator is required for creating representative corpuses of data for teaching automated learning algorithms for predictive maintenance.

Flexible assembly system was described by Donath and Graves [1] as a system consisting of a set of products each with a specified volume assembled on a workshop consisting of a fixed number of cells. In practise, a flexible assembly system contains parts and materials stored in an intermediate storage, a conveyor or crane system to move the parts, materials, intermediate assemblies and finished products between the cells, and the cells with work machines and necessary tooling to assemble and process intermediate assemblies and products. The cells might consist of for example manual assembly steps, robotic assembly cells, CNC lathes, or 3D printing.

As opposed to more specialized industrial production systems, flexible assembly systems are designed for smaller batches and greater flexibility so that the set of end products can vary more widely. For example, flexible assembly systems can be reconfigured more conveniently following the evolution of new versions of the end products. Flexibly assembly systems also allow for a wider range of customization between the different end product instances of the same product. The nature of fast evolution of configurations and workflows, and heterogeneous operating conditions make planning maintenance more challenging.

System downtime is a significant cost for flexible assembly systems. System downtime is reduced by preventive maintenance typically scheduled periodically. Recently Internet of Things and opening of the networked industrial system APIs have created new possibilities for predictive maintenance. Different sites with similar flexible automation systems have widely different environmental and workload conditions which has an impact on the wear and tear of the flexible automation system components. Taking different site environments and workloads into account when planning preventive maintenance cycles is non-trivial. There is a clear need to adapt maintenance based on actual conditions in the operation rather than by simply scheduling maintenance periodically.

Going from preventive maintenance towards predictive maintenance optimizes and targets the maintenance related costs towards the activities that have best impact on improving the availability and operation of deployed flexible manufacturing systems. Predictive maintenance in flexible assembly systems benefits from automatic indicators for potential future faults. Current predictive maintenance systems concentrate on measuring device health by means of measuring temperature and vibration [2].

Creating novel automatic methods for fault prediction requires standard benchmarks to compare these methods against. Currently there are no standard benchmarks to test different anomaly detection methods and compare their performance against each other. Such standard benchmarks exist in other fields of machine learning, such as the MNIST handwritten digits benchmark [3]. Realistic simulations are often used [4] for domains where real data is hard to collect or where real data would be subject to business confidentiality. Training neural network and other learning systems often utilizes realistic simulations instead of real data. Real measured data is often limited in both state space and in volume, and often simulations give better results [5].

Fault-free simulations of flexible assembly systems are generally available [6] and have been studied for visualization and optimization purposes. These are not directly suitable for benchmarking fault detection methods.

II. Problem Formulation

A standard benchmark should reflect a realistic task and it should have enough data to be useful.

Real data regarding flexible assembly system faults is not generally available, because fault data is generally subject to business confidentiality in flexible assembly industry. The generally available simulation models do not typically contain failure modes and are not typically designed to emit log structured events in a realistic fashion, but they can be used as a starting point in creating a flexible assembly system model.

To enable relevant research into anomaly detection in flexible assembly systems a realistic simulator for log structured events is needed. All references to simulated faults and failure modes in the research described in this article are hypothetical and should not be considered to reflect actual characteristics of any specific existing systems. An effort is made to make the simulations contain the necessary features of generic and realistic failure modes to make the models useful for research, but for example the simulated fault frequencies and the configuration of the flexible assembly system along with workflows and end products are arbitrary and hypothetical.

A simulation for benchmarking anomaly and fault detection methods should not be fully deterministic to reflect the real world dynamics. The simulation should also include realistic

faults with possible early indicators such as variations in delays in the steps of the process.

III. Modes of Failures in Flexible Assembly Systems

Flexible assembly system failures are typically component failures where a failure of a single module or a component will lead to the system becoming inoperational. Since the flexible assembly systems are dynamic systems with numerous moving parts, the wear and tear of the components and tools are a significant source of downtime. In addition to the component failures, certain systemic failures can cause downtime, such as power loss and network communication failures.

The systemic failures mentioned don't typically show any indication before they happen so predictive maintenance has little potential to be applied there. Tool wear and tear depends on the workload. Typically tools that wear out fast, for example multiple times for one unit of work in a cell, receive much attention and such tools are generally well maintained. Tools that have wear and tear but have longer life spans are more susceptible to being overlooked by operators and might have some indicators of degrading before a failure. Cranes, hatches and conveyors have relatively long life spans and are primarily maintained in periodic preventive maintenance. If we could get indicators for impeding failure for these components, the periodical maintenance could be scheduled earlier to prevent downtime. In addition to these physical faults the flexible assembly system can suffer from human errors. Human operator can inadvertently misconfigure the flexible assembly system so that its operating mode changes unexpectedly, for example by disabling a cell which can lead to the system stopping without a proper failure. In addition to configuration errors, human operators might fail in manual assembly steps for example by pressing the button to mark the step as completed too quickly. The flexible assembly system might also get in incompatible materials, parts or replacement tools and fail when trying to use them.

There has lately been an increase of computer crime against industrial networks as the devices and systems become more connected. Nation states execute industrial espionage and even sabotage against each other [7]. An electronic sabotage attack might present itself much like a human misconfiguration error, and would be potentially detected in a similar fashion even if the attack itself would not be directly visible in the logs as anomalous activity.

For useful anomaly detection systems we need to concentrate on faults that are unexpected, and that potentially have early indications. The table I summarizes the classes of faults.

TABLE I
Table of recognized fault types in flexible assembly systems

Fault type	Unexpected faults	Potentially has an early indication	Useful target for anomaly detection
Auxilliary component wear and tear	X	X	X
Human error, misconfiguration / Cyberattack	X	X	X
Human error, failed manual step	X	X	X
Incompatible parts, materials or replacement tools	X		
Systemic failures (power, network)	X		
Tool wear and tear for quickly degrading tools		X	
Tool wear and tear for slowly degrading tools	X	X	X

IV. Flexible Assembly System Model

The simulation consists of a realistic model of a flexible assembly system with a class of faults randomly injected into the process. The simulation will not be fully deterministic and contains soft faults which do not affect operation in addition to hard faults which result in immediate downtime.

The simulator is implemented in Python and it generates a JSON file which models the logs from the system. The log message fields are defined in Table **??** and and example message is shown in **??**.

The simulation framework consists of a scheduler that takes the next event from the event queue and applies it to the respective module. The simulated modules execute when events are applied to them and generate new events and immediate log entries with timestamps and other metadata. The simulated modules have optional active failure modes that affect the simulated operation of the component through generated log messages or the subsequent generated events. Failure modes are set up to the simulated modules in a separate failure module. The overall framework sets up the simulation and events.

A. Normal Operation of the Simulated System

The simulated hypothetical system assembles and tools a car transmission block loosely inspired by a Youtube video of Chrysler transmission assembly [8]. The workflow consists of several manual assembly steps in separate cells and transporting the subassemblies between the cells by the means of cranes and conveyors. Multiple transmission blocks are being assembled simultaneously in separate cells.

The transmission component consists of four subassemblies. The main sequence starts from the frame of the transmission block and continues through several manual steps done in separate cells. The steps of the main sequence are given in the Tables II and III. Each step can contain a separate sequence of events and associated log messages. The mean durations for manual steps can vary in normal operation $\pm$ thirty percent. The mean durations for automatic steps can vary $\pm$ five percent.

The production frequency of the transmission blocks depends on the slowest step as in this case there is only one cell for one step and no parallel work is possible within one step. The slowest step takes about 76 seconds. In this

TABLE II
TABLE OF THE STEPS 1-16 IN THE MAIN ASSEMBLY PROCESS

Step	Description	Duration	Log messages	Potential failures
1	Crane	30 s	Going forward, stopping, going back, stopping	Wear & tear
2	Manual inspection	37 s	OK pressed	Failed manual step
3	Conveyor	30 s	To cell, stop, to next cell	Wear & tear
4	Bowl feeder gives components	5 s	Given	Wear & tear
5	Add components	21 s	OK pressed	Failed manual step
6	Conveyor	30 s	To cell, stop, to next cell	Wear & tear
7	Bowl feeder gives components	10 s	Given	Wear & tear
8	Add components	34 s	OK pressed	Failed manual step
9	Conveyor	30 s	To cell, stop, to next cell	Wear & tear
10	Crane with subassembly A	10 s	Going forward, stopping, going back, stopping	Wear & tear
11	Combine with subassembly A	34 s	OK pressed	Failed manual step
12	Conveyor	30 s	To cell, stop, to next cell	Wear & tear
13	Conveyor with subassembly B	10 s	To cell, stop	Wear & tear
14	Combine with subassembly B	35 s	OK pressed	Failed manual step
15	Conveyor	30 s	To cell, stop, to next cell	Wear & tear
16	Bowl feeder gives components	5 s	Given	Wear & tear

TABLE III
TABLE OF THE STEPS 17-31 IN THE MAIN ASSEMBLY PROCESS

Step	Description	Duration	Log messages	Potential failures
17	Conveyor with cover	10 s	To cell, stop	Wear & tear
18	Add cover and bolts	76 s	OK pressed	Failed manual step
19	Conveyor	30 s	To cell, stop, to next cell	Wear & tear
20	Tighten the bolts	28 s	OK pressed	Failed manual step
21	Conveyor	30 s	To cell, stop, to next cell	Wear & tear
22	Conveyor with subassembly C	10 s	To cell, stop	Wear & tear
23	Combine with subassembly C	60 s	OK pressed	Failed manual step
24	Conveyor	21 s	To cell, stop, to next cell	Wear & tear
25	Tighten the bolts	16 s	OK pressed	Failed manual step
26	Conveyor	21 s	To cell, stop, to next cell	Wear & tear
27	Bowl feeder gives components	5 s	Given	Wear & tear
28	Add components	11 s	OK pressed	Failed manual step
29	Conveyor	21 s	To cell, stop, to next cell	Wear & tear
30	Tighten the bolts	32 s	OK pressed	Failed manual step
31	Conveyor	21 s	To output gate	Wear & tear

hypothetical assembly plant, new transmission block frames are added every 100 seconds and the final transmission blocks are being produced roughly in the same interval. This means that there will be at maximum roughly seven transmission blocks in different stages being produced at the same time.

B. Simulated Failure Modes

It is regrettable that real data on faults in real flexible assembly systems are not available, but in any case it is possible to make reasonable models based on existing data from other analogous sources such as [9] and [10].

The previous subsection describes the normal operation of the system. The failure modes of the manual steps include human errors by pressing OK prematurely without properly executing the step. This leads to the next manual step failing and returning the block back to the previous step or removing it from the assembly line altogether. Misconfiguration can cause different kinds of results, but here we assume some work step is skipped so that it is not noticed and doesn't directly prevent other steps from being executed. Failed manual steps also include wear and tear on the tools which gradually increases the duration of the work before causing a fault. The human operator can also tire and lose concentration when doing lots of repetitive steps. There are studies showing how speed performance degrades in general in a linear fashion in humans under fatigue and boredom, for example [11]. Switching human operators during the day might change the effective performance slightly.

The automatic steps wear and tear cause gradually increasing durations for the step also until at some point the step stops functioning and intermediate assemblies start piling up until a fault is signalled because of exceeding the limits of intermediate storage space.

Duration increases before a fault are non-deterministic and for some faults they don't actually happen before the fault. The profile of the duration increase depends on the type of device used for the step. Generally mechanical wear and tear effect on machine health profile follows an accelerating curve downwards [12], [13]. For the purposes of simulation I assume machine health has an approximately linear effect on

machine performance measured in delays and success rates in applicable types of faults.

Simulated wear and tear causes two types of delay profiles. First, a continuous wear and tear typically causes no measurable delays at first, but when the degradation accumulates the delay increases following a roughly exponential curve **??**. Second, for certain steps, such as taking bolts from the bowl feeder a human operator might fail to get a screw from time to time. Typically human operator tries again and succeeds. The frequency of the failure increases slowly and roughly following an exponential curve until the failure rate becomes apparent and causes the operator to take corrective action, typically causing downtime. These failures cause a random delay if they happen. This characteristic delay profile is depicted in **??**.

While the directly visible fault modes are easier to model and characterise, complex flexible assembly systems also have warnings and trace messages which are signalled into logs, but are not signalled to operators in real time because most of the time these messages are not a cause for concern. An example of such a signal could be for example a health warning of a device that happens spuriously once every now and then for normal operation, but increases in frequency when the device actually starts to fail. There can also be log messages for example for electrical switches turning on and off when the working step is active, but never while the step is not active. Anomalies in such messages are an indicator of potential future fault.

V. Evaluation

The baseline benchmark is done by automatically inferring a deterministic finite automaton from the fault-free log events as was shown by Langer et al. [14]. This trivial model does not work very well, but can be used as a baseline for more advanced methods. The finite automaton is then used to evaluate different logs generated to contain certain faults.

VI. Results

VII. Conclusion

This study presents a benchmark model for validating anomaly detection methods for flexible assembly systems using log structured data. Existing systems for predictive maintenance concentrate on measuring health of a separate devices based on typical degradation characteristics for such devices. They also concentrate on continuous measurement data instead of log entries.

The diagnostic data for flexible assembly systems is a closely guarded secret for business reasons, but it is possible to make reasonable models of faults based on other analogous sources.

The flexible assembly system simulator, FAS Simulator, is published in GitHub [15] as open source. This kind of simulator would have little realism, and in fact little use unless it was continuously developed in dialogue with real flexible assembly systems. While the simulation approach adds an indirection between real data and developed methods, it is necessary to veil the business critical real fault and diagnostic data of the fast assembly systems and to make the different methods comparable. The simulator also provides means to incorporate new types of failures and test anomaly detection methods in a quick iteration.

Additional research is needed to improve this benchmark simulation by using experiences from real flexible manufacturing systems and their modes of failure and respective future fault indicators. For example, continuous measurement values from temperature and vibration sensors would be good targets to simulate. I will be continuing this work in the future in association to my future research conserning automated anomaly detection in flexible assembly systems.

References

[1] M. Donath and R. J. GRAVES, "Flexible assembly systems: an approach for near real-time scheduling and routeing of multiple products," *The International Journal Of Production Research*, vol. 26, no. 12, pp. 1903–1919, 1988.

[2] L. R. Contreras, C. Modi, and A. Pennathur, "Integrating simulation modeling and equipment condition diagnostics for predictive maintenance strategies-a case study," in *Simulation Conference, 2002. Proceedings of the Winter*, vol. 2. IEEE, 2002, pp. 1289–1296.

[3] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[4] G. Jäger, S. Zug, T. Brade, A. Dietrich, C. Steup, C. Moewes, and A.-M. Cretu, "Assessing neural networks for sensor fault detection," in *Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 70–75.

[5] W. Duch, *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005: 15th International Conference, Warsaw, Poland, September 11-15, 2005, Proceedings*. Springer Science & Business Media, 2005.

[6] G. Rosati, M. Faccio, C. Finetto, and A. Carli, "Modelling and optimization of fully flexible assembly systems (f-fas)," *Assembly Automation*, vol. 33, no. 2, pp. 165–174, 2013. [Online]. Available: http://dx.doi.org/10.1108/01445151311306690

[7] D. Kushner, "The real story of stuxnet," *IEEE Spectrum*, 2013. [Online]. Available: http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet

[8] Cars, "Chrysler transmission assembly line," 2014. [Online]. Available: https://www.youtube.com/watch?v=447sSED91v0

[9] "Nasa ames prognostics center of excellence - data repository." [Online]. Available: http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/

[10] P. H. Tsarouhas, "Classification and calculation of primary failure modes in bread production line," *Reliability Engineering & System Safety*, vol. 94, no. 2, pp. 551–557, 2009.

[11] C. S. Pan, R. L. Shell, and L. M. Schleifer, "Performance variability as an indicator of fatigue and boredom effects in a vdt data-entry task," *International Journal of Human-Computer Interaction*, vol. 6, no. 1, pp. 37–45, 1994. [Online]. Available: http://dx.doi.org/10.1080/10447319409526082

[12] O. Eker, F. Camci, and I. Jennions, "Major challenges in prognostics: Study on benchmarking prognostics datasets," *PHM Europe, 6th–8th July*, 2012.

[13] K. G. N. A. . A. A. U. Berkeley), "Documentation for mill data set." [Online]. Available: http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/

[14] F. Langer, K. Bertulies, and F. Hoffmann, "Self learning anomaly detection for embedded safety critical systems," *Schriftenreihe des Instituts für Angewandte Informatik, Automatisierungstechnik am Karlsruher Institut für Technologie. KIT Scientific Publishing*, pp. 31–45, 2011.

[15] T. Keski-Valkama, "Flexible assembly simulation: FAS Simulator," https://github.com/keskival/FAS-Simulator.

Tero Keski-Valkama Tero Keski-Valkama is working as a software architect in Cybercom Finland Oy. He has been programming neural networks since high school.