Detecting Anomalous Energy Consumptions in Distributed Manufacturing Systems

Sebastian Faltinski[1], Holger Flatt[1], Florian Pethig[1], Björn Kroll[1], Asmir Vodenčarević[2], Alexander Maier[3] and Oliver Niggemann[1,3]
[1]Fraunhofer IOSB-INA, Application Center Industrial Automation
[2]University of Paderborn, Knowledge-Based Systems Research Group
[3]Ostwestfalen-Lippe University of Applied Science, inIT-Institute Industrial IT
Email: {sebastian.faltinski, holger.flatt, florian.pethig, bjoern.kroll, oliver.niggemann}@iosb-ina.fraunhofer.de, alexander.maier@hs-owl.de, asmir.vodencarevic@upb.de

***Abstract*—This paper presents a novel model-based approach for the prediction of energy consumption in production plants in order to detect anomalies. A special Ethernet-based data acquisition approach is implemented that features real-time sampling of process and energy data. Hybrid timed automaton models of the supervised production plant are generated and executed in parallel to the system by using data samples as model input. According to comparisons of predicted energy consumption with the production plant observations, anomalies can be detected automatically.**

An evaluation within a small factory shows that anomalies of 10 % differences in energy consumption, wrong control sequences and wrong timings can be detected with a minimum accuracy of 98 %. With this approach, downtimes of production systems can be shortened and atypical energy consumptions can be detected and adjusted to optimal operation.

I. Introduction

The automatic detection and optimization of suboptimal energy consumption in production plants is a key challenge for European industry within the next years [1]. This is mainly due to the fact that energy prices have increased in the past and will still raise in the future. Another reason for a greener production is driven by political goals. E.g. in Germany, taxes on electricity for production related companies will raise in 2013 as long as no energy management system (e.g. according to standards such as ISO50001) is installed.

Currently, operators are mainly concerned about a plant's correct functionality; energy aspects are still regarded as of secondary importance. And even if energy aspects are seen as a central issue, operators mainly replace single hardware components, e.g. the installation of an energy-optimized drive.

A large optimization potential arises from system level: plant modules, engines and drives should be controlled in a way that takes energy aspects in consideration. E.g. synchronized starts of drives should be avoided since they can cause expensive harmonics. Furthermore, errors leading to suboptimal energy consumption in a plant can be detected and repaired, e.g. blocked pipes or worn drives [2].

A first step towards an industrial real-time energy management system is the anomaly detection. In general, two algorithmic approaches exist for detection of anomalous or suboptimal plant situations:

Phenomenological Approach: The system's output including its energy consumption is directly classified as correct or anomalous. In such approaches, the classifier (e.g. k-nearest neighbour) is trained with learning patterns. In the following classification phase anomalies are detected by calculating the membership to one of the learned classes (good or anomalous) i.e. anomalies are detected.

Model-based Approach: In order to detect suboptimal energy situations automatically, a model-based approach as depicted in figure 1 can be used. A model is used to predict the normal energy consumption of a plant. For this, the simulation model needs all the inputs of the plant, e.g. product information, plant configuration and plant status. If actual energy measurements vary significantly from simulation results, the energy consumption is classified as anomalous.

While phenomenological approaches are often more straight-forward and do not require a model, they have one major inherent drawback: They must deduce again the direction of causality since they deduce from measurements (i.e. symptoms) to anomalies. For complex distributed systems with many interdependencies between components and complex causalities, this leads to several problems: (*i*) The classification rules need a high number of measurement variables – including the measurements' history – to differentiate between error causes. (*ii*) A high number of classification rules are needed to capture the effect of different input combinations and again

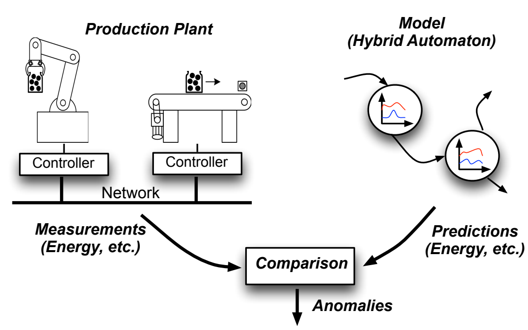


Fig. 1. Model-based Anomaly Detection Approach

all combinations over time may be needed.

Since the analysis of a plant's energy consumption depends on a high number of input signals and normally deals with distributed plants and automation systems, a model-based approach is chosen here.

This paper presents a novel approach for model-based anomaly detection of energy consumptions in production plants. First, the current state of the art is reviewed in section II. A special data acquisition approach is developed in section III. Next, a suitable formalism for modeling energy consumption is issued (section IV). According to the formalism, the models are learned based on plant observations (section V). For this purpose, the HyBUTLA algorithm is described. Results of an anomaly detection case study are given in section VI. Finally, section VII concludes this paper.

II. State of the Art

The following section reviews the state of the art of the three main issues of this paper: data acquisition, energy model generation and anomaly detection.

Data Acquisition: Before energy data can be analyzed, real-time measurements have to be acquired from field level. Energy control functions, integrated into common manufacturing execution system (MES) applications [3], [4], [5] are not suitable due to their focus on assigning energy costs to specific fabrication units.

In order to measure energy and capture process data within shorter intervals, the programmable logic controller (PLC) can be tapped by using classic OPC (OLE for Process Control) [6] or the advanced OPC-UA (OPC Unified Architecture) [6]. This technique is mainly focused on a distributed network with several PLCs monitored by human machine interface (HMI) and MES applications. The data analysis with the objective of building an energy consumption model requires a system wide precise time-stamp. Due to unsynchronized PLCs and other delays, e.g. processing times, OPC-based data acquisition is not suitable for obtaining real-time data.

Special energy profiles have been added to real-time Ethernet protocols like PROFINET (PROFIenergy) and SERCOS (SERCOS Energy). These profiles include definitions for the transmission of energy data via production networks. Data acquisition approaches fixed to these profiles cannot be flexibly integrated into heterogeneous automation systems, where data has to be captured from a variety of bus systems.

Energy Model Generation: Currently the research focus is on off-line energy behavior modeling. As part of [7], the energy consumption of machine tools and components in dependence on their specific process requirements are quantified by simulation. The results are exclusively used for product design during the construction phase. In [2] an energy behavior model is generated in order to find an energy aware usage profile by simulation.

On-line energy consumption models that can be executed in real-time and parallel to the process with the aim of anomaly detection have not been researched so far.

Anomaly Detection: Anomalies can be detected manually by observing the acquired energy data. Based on this approach, in [8] a software architecture for data monitoring and analysis is developed. Model-based approaches support automatic detection of anomalies within production processes. Instead of creating an energy consumption model, a petri net based event model is generated in [9]. As a limitation, the approach only covers anomalies which result in an event change.

Overall, recent approaches are mainly limited to data-acquisition and related tasks. The creation of an energy behavior model with further application for anomaly detection has not been published so far. This paper overcomes this gap by including these steps into one system approach in order to detect anomalies in production plants.

III. Data Acquisition

Process and energy data acquisition in distributed systems like manufacturing plants is a challenging task. This is mainly due to the three following facts:

Heterogeneous Automation Systems: Within a plant, different automation and measuring systems are used. Devices from different vendors, protocols and interfaces coexist and data has to be integrated by using technologies like Classic OPC [6] or the OPC Unified Architecture (DIN EN 62541) [6]. Although OPC is widely used, it cannot cope with the essential requirement for diagnosis in high-speed applications like motion control: Acquiring synchronized data with an appropriate accuracy.

Timing Requirements and Synchronized Measurements: Production plants consist of several assets which again consist of a multitude of machinery. An ideal data acquisition solution would make it possible to acquire process data in the needed quality, i.e. synchronized and with an acceptable accuracy in time over different production processes. Only data acquired in such a way can be used to analyze a system's status. Since modern automation systems can handle cycle times in the sub-millisecond range, mentioned solutions like OPC are not sufficient. For diagnosis the timestamp has to be generated as close to the signal's source as possible whereas current approaches are generating timestamps in the PLC.

Data Integration: The third issue regarding data acquisition results from data integration challenges. Existing data acquisition solutions often use proprietary interfaces preventing an easy data access. Here, OPC UA is helpful, since it provides a web service interface allowing for world wide data access. Furthermore, it provides an explicit meta-model for describing the semantic of acquired signals. This is a prerequisite for the data analysis, since complex information has to be machine-sensible if analyzed automatically [6].

A new Datalogger Architecture: Currently, there is no standard for future smart meters measuring the energy consumption of smart homes and smart factories. By using meters existing today, one has to deal with the prementioned challenges. One approach to overcome these issues is presented next: A datalogger architecture for distributed Ethernet networks. Ethernet frames can be captured and timestamped with an accuracy of approximately 10 ns and additionally multiple datalogger devices can be synchronized with an accuracy of $< 10\ \mu s$ by using the IEEE 1588 Precision Time Protocol (PTP) [10]. Data access has been realized using an OPC UA Server. Using this setup, process and energy data transferred via Ethernet-based protocols, e.g. PROFINET or Modbus/TCP can be logged with very high accuracy. Figure 2 shows the implemented data acquisition architecture.

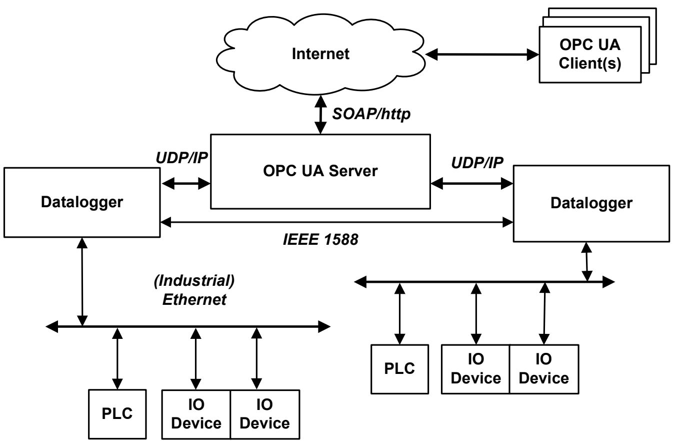


Fig. 2. Architecture of the Data Acquisition Approach

Visualization: Using mobile devices (e.g. smart phones) for visualization of the energy consumption model is a suited approach since they are widely spread across production environments. Their modern architecture allows powerful data processing as well as user-friendly presentation. These features were utilized in order to implement a tool presenting warnings or anomalies immediately. To achieve this objective, an OPC UA server is utilized for providing data values combined with site-specific information.

IV. Model Formalism

After energy measures have been acquired from the field communication, this data is used to build a suitable model. Here, a suitable behavior model formalism is needed to represent the energy consumption in a way that can be used for anomaly detection later on.

A. Requirements to the Formalism

The following requirements must be fulfilled by a modeling formalism for the energy consumption of production plants:

Time Modeling: Since energy consumptions are functions over time, a formalism must at least partially capture this time dependency. To allow for an automatic time analysis, time should be modeled explicitly. An important parameter is the required resolution of time: Higher resolutions predict a system's behavior more precisely, but require very detailed models that often cannot be learned.

Dependency Modeling: The energy consumption model must deduce a plant's energy consumption, which is the model's output, based on input values that describe the current plant status.

Input Values: Actuator and sensor values, MES settings such as the current product specification, ERP values such as raw material specifications, plant descriptions such as physical models and automation system specifications.

Output Values: Active and reactive power, harmonics.

Therefore, the model formalism must be able to compute the output values (dependent variables) based on the input values (independent variables) and must model all dependencies between these variables.

Mode Modeling: A system may, at some point in time, fundamentally change its behavior and therefore also its energy consumption. E.g. a vehicle changes its gear or a valve in a chemical reactor is opened. These disruptive changes correspond to different system modes. A mode is a discontinuity of the system, a point in time at which the model must be changed significantly, because the system behavior changes significantly [11]. The formalism must be able to model multiple system modes.

Hybrid Modeling: Input and output variables can be discrete (e.g. binary) or may have continuous values. Binary variables often model events, which cause a mode change. The formalism must model such hybrid systems.

Learnability: A manual model creation for each plant is often not possible due to insufficient expert knowledge and often not feasible because of the lack of time. Therefore, plant models should be learned: Based on observations, the model is abstracted automatically. In some cases only some parameters of the models are learned.

Asynchronism Modeling: Often, some plant parts operate asynchronously, i.e. their relative timing does not follow a specific pattern and can therefore not be predicted. Here, only synchronous systems are regarded. If system parts behave asynchronously, they must be modeled and analyzed separately.

Based on these requirements, a two-step modeling approach is implemented:

I. Structure Model: The structure model defines the system as a set of interconnected communicating components. Components may run asynchronously, i.e. these models capture aspects such as asynchronism and dependencies. Such models are well-known and are described in the following section IV-B.

II. Behavior Model: For each component, the behavior must be modeled. To this aim, a hybrid timed automaton formalism is given in section IV-C.

B. Structure Model

From a model learning perspective, a structure model subdivides the overall system into components working in parallel. I.e. a structure model defines a (hierarchical) set of interconnected components where components work in parallel (usually asynchronously) and each individual component shows a sequential behavior only. In plants, such sequential components often correspond to one technical device such as a robot, a conveyor belt, a reactor or a PLC.

Using sequential components is a prerequisite for both, manual and automatic model generation, because otherwise a component's behavior model will become too large and too confusing. If a behavior model tries to capture a parallel behavior of several components by means of a sequential behavior model, a combinatorial growth of the number of model states will result.

The reader may note that no approaches for learning structure models exist and we therefore have to model them manually. In the following, a structure model and its components are defined formally; the definition here is especially tailored for the purpose of model learning.

Definition 1 *(Component) A component C is defined by a behavior function $b_C : \mathbf{R}^{m+1} \rightarrow \mathbf{R}^n, n, m \in \mathbf{N}$. b_C is a function over m input variables and over time and it returns n output variables [12].*

A structure model is created by connecting several components:

Definition 2 *(Structure Model) A structure model M is defined as a tuple $< \mathcal{C}, z >$ where $\mathcal{C} = \{C_0, \ldots, C_{p-1}\}$ is the set of components and $z : \mathcal{C} \times \mathbf{N} \rightarrow \mathcal{C} \times \mathbf{N}$ maps an output variable of one component onto the input variable of another component.*

I.e. $z(C_i, k) = (C_j, l)$ connects the k's output variable of C_i with the l's input variable of C_j.

C. Behavior Model

Now, for each component a behavior model, defining b_C, is needed. In general, a large variety of such models exists, e.g. differential equations or finite state machines. Table I shows some typical modeling formalisms and assesses them according to the requirements from above. Based on this analysis, the hybrid timed automata evolve to be a suitable approach.

Figure 3 shows an example of a hybrid timed automaton.

A transition between two states *run* and *stop* is triggered by an event a, this often corresponds to a mode change of the system. The transition is only taken if a timing constraint $t < 20\ ms$ holds. Additionally, a probability $p = 0.2$ for taking the transition is given. Within a state, continuous signals may change according a function θ.

In the following, a hybrid timed automaton is defined:

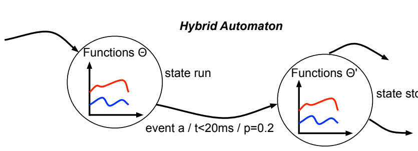


Fig. 3. Example of a Hybrid Automaton

Definition 3 *(Hybrid Timed Automaton) [12] A hybrid timed automaton is a tuple $A = (S, s_0, F, \Sigma, T, \Delta, Num, c, \Theta)$, where*

- *S is a finite set of states, $s_0 \in S$ is the initial state, and $F \subseteq S$ is a set of final states,*
- *Σ is the alphabet comprising all relevant events.*
- *$T \subseteq S \times \Sigma \times S$ gives the set of transitions. E.g. for a transition $\langle s, a, s' \rangle$, $s, s' \in S$ are the source and destination states and $a \in \Sigma$ is the trigger event.*
- *A set of transition timing constraints Δ with $\delta : T \rightarrow I, \delta \in \Delta$, where I is the set of time intervals.*
- *A function $Num : T \rightarrow \mathbf{N}$ counts the number of observations that used the transition.*
- *A single clock c is used to record the time evolution. At each transition, the clock is reset. This allows only for the modeling of relative time steps.*
- *A set of functions Θ with elements $\theta_s : \mathbf{R}^n \rightarrow \mathbf{R}^m, \forall s \in S, n, m \in \mathbf{N}$. I.e. θ_s is the function computing signal value changes within a single state s.*

In contrast to traditional automata, the formalism is simplified here to ease the learning task: First of all, only relative timing is allowed, e.g. transitions may not refer to a global time base. Furthermore, transition probabilities are added and arbitrary function models are supported to describe value changes in states, instead of differential equations only.

V. Learning of Energy Models

In order to learn the overall model, first the structure model (see section IV) has to be defined and then the behavior models have to be learned for each component individually. Since no general algorithm exists to derive a structure model from system measurements, the implemented approach relies on manual creation of structure models or on domain specific methods.

As outlined in detail in section II, within the domain of production plants some constraints have to be considered: (*i*) Besides transitions, states must also be learned. (*ii*) Only positive examples exist (i.e. measurements). (*iii*) Discrete and continuous signals must be modeled. (*iv*) Timing and probability information must be modeled. Further, the HyBUTLA learning algorithm [20] is described shortly (see also figure 4):

First, in **step (0)** all relevant signals are measured during a system's normal operation and are stored in a database.

In **step (1)**, events are generated; events usually correspond to mode switches. An event is generated whenever a variable changes its value – this corresponds to an actuator or sensor signal in a technical system such as switching a valve or the toggling of a photoelectric barrier.

Often, such events appear several times in the original observations. In this case, it is checked whether the events belong to the same process; this is done based on the events' timing. Therefore, the probability density function (PDF) – probability over time – is computed for each available event a.

TABLE I
ANALYSIS OF DIFFERENT BEHAVIOR MODELING FORMALISMS.

	Time	Dependencies	Modes	Hybrid	Learnability
Function Approximation e.g. Neural Network, Support Vector Machines	No	Yes	No	Discrete variables are treated as continuous variables	Yes
Differential Equations	Implicitely	Yes	No (only with additional event handling [13])	Discrete variables are treated as continuous variables	Parameters can be learned [14], the equations themselves can not be learned in most cases [15]
Decision Trees, Rules	No (partially in [16])	Yes	Yes	No	Yes, e.g. via ID3 [17]
Timed Automata	Explicitely	Yes	Yes	No	Yes, e.g. in [18], [19]
Hybrid Timed Automata	Explicitely	Yes	Yes	Yes	Yes, see section V

If the PDF is the sum of several distributions, separate events are created for each distribution.

In **step (2)**, a prefix tree acceptor (PTA) is built. Such a PTA is a hybrid timed automaton in the form of a tree. In a PTA, each sequence of observations results in one path from the tree's root to a leaf; common prefixes of sequences are therefore stored only once.

The continuous behavior (i.e. the output process variables) within each state $s \in S$ is modeled by learning the functions $\theta_s \in \Theta$. To this aim, machine learning methods such as linear regression or neural networks are applied using the portion of recorded sensor measurements associated with the corresponding state.

Now, in **step (3)** compatible states are merged until a reduced automaton is reached, which can still predict the system behavior: First, the compatibility between two states is checked in the bottom-up order. The order of the nodes is defined by the lexicographic order of the shortest sequence of events leading to the node. The bottom-up merging order means a significant speed-up of the algorithm for most datasets compared to the top-down strategy [21], [19].

If states are found to be similar, they are merged. The compatibility criterion consists of several similarity tests. If two states are merged, their portions of observed data are combined and new functions θ are learned. Because after the merging step the resulting automaton may be non-deterministic, the sub-trees of the new merged state are made deterministic by merging their nodes recursively (see also [22] for details).

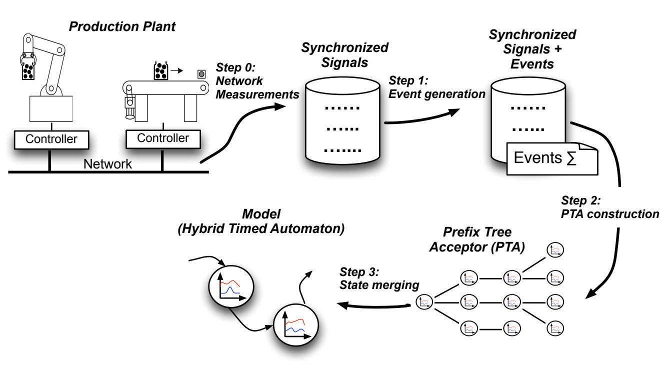


Fig. 4. General Concept for Learning Hybrid Timed Automata

VI. ANOMALY DETECTION

The goal of anomaly detection in general is to find anomalous objects caused by faulty operating conditions in a system. According to this aim, automatically learned behavior models are used by the anomaly detection algorithm ANODA [12], which can detect anomalies in both continuous and discrete parts of the system. It targets three types of anomalies:

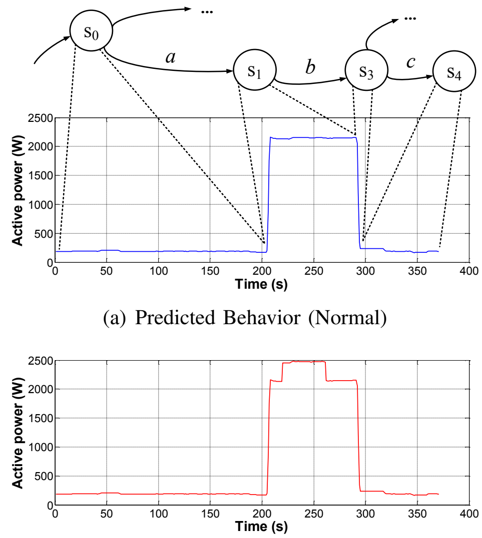
Anomalous Energy Consumption: Energy over- or underconsumption is detected by comparing model predictions with system observations. Figure 5(a) shows a real-world power signal predicted by a learned hybrid automaton. Observed behavior is given in figure 5(b), where a discrepancy (overconsumption) is detected in state s_1 of an automaton.

Control Sequence Anomalies: Failures in the control system often lead to unknown operating conditions in the plant. Detecting such wrong sequences of control (discrete) signals is a prerequisite for energy efficiency.

Timing Anomalies: Even when the sequence of control signals is correct, the anomaly can occur in their timing (e.g. due to frazzled valves) causing suboptimal operation.

The experiments were conducted in a real, small plant used for storing, transporting, processing and packing bulk material, called the 'Lemgo Model Factory'. The HyBUTLA algorithm [20] was used with 12 production cycles (average length of 191 samples) with 6 continuous and 6 discrete signals to learn the popping machine model with 19 states (prefix tree acceptor (PTA) had 52). The monitored signal for this component is its active power. It is shown in figure 5.

Around 20% of the samples in the test production cycle were anomalous. Targeted anomalies are: energy underconsumption by 10% (a_1), energy overconsumption by 10% (a_2), wrong control sequence (a_3), and timing anomaly (a_4). Tracked performance metrics include sensitivity (true positive rate), specificity (true negative rate), and the overall accuracy. The experiment was repeated 100 times, and its summarized results are given in table II. The scalability of this approach was evaluated on the artificial dataset, around 15 times larger than the real one (12 cycles of average length 818 samples,



(a) Predicted Behavior (Normal)

(b) Observed Behavior (Overconsumption)

Fig. 5. Real-world example of the anomalous energy consumption

with 21 continuous and 20 discrete signals). The learned model had 59 states (in contrast to PTA's 151). The same anomalies were targeted and the results, averaged over 100 experiments, are shown in table III.

TABLE II
PERFORMANCE METRICS FOR THE REAL SYSTEM

Performance	Anomalies			
	a_1	a_2	a_3	a_4
Sensitivity(%)	94.74	99.32	100	100
Specificity(%)	100	98.86	100	100
Accuracy(%)	98.93	98.95	100	100

TABLE III
PERFORMANCE METRICS FOR THE ARTIFICIAL DATA

Performance	Anomalies			
	a_1	a_2	a_3	a_4
Sensitivity(%)	67.39	64.77	100	100
Specificity(%)	94.60	94.66	100	100
Accuracy(%)	89.13	88.66	100	100

VII. CONCLUSION

This paper proposes a new approach for acquiring time synchronized energy measurements in a distributed production system in order to build an energy consumption model of the entire system. Based on this model, anomalies in the energy consumption of plants can be detected. This supports advanced condition monitoring and helps to detect abnormal behavior of certain components automatically. For this prediction system, a model based on a hybrid timed automaton is learned with small portion of expert knowledge. The approach is able to detect abnormal energy consumption, control sequence anomalies and timing anomalies.

Future work will be focused on energy efficient and factory-wide control, which is based on the energy consumption forecast derived from the learned model. Therefore, the proposed solution will support both, the detection of abnormal energy consumption and the reduction of total energy costs.

REFERENCES

[1] H.-L. Group, "MANUFUTURE - Strategic Research Agenda," European Commission, Tech. Rep., 2006.
[2] A. Dietmair and A. Verl, "Energy consumption modeling and optimization for production machines," in *Sustainable Energy Technologies, 2008. ICSET 2008. IEEE International Conference on*, 2008, pp. 574 –579.
[3] Siemens, "Energy management cuts factory costs," Innovation News Document IN 2009.08.4, 2009.
[4] Mitsubishi, "Energy control pack, shop floor energy management," Tech. Rep. Document EBG 218-EN, 2010.
[5] ABB Automation GmbH, "Energy management solution for the process industry, cpmplus energy manager," Tech. Rep. Document 3BFI 401001 R0101, 2009.
[6] M. Damm, S. Leitner, and W. Mahnke, *OPC Unified Architecture*. Springer-Verlag Berlin Heidelberg, 2009.
[7] e SimPro, "Effiziente Produktionsmaschinen durch Simulation in der Entwicklung," www.esimpro.de.
[8] A. Vijayaraghavan and D. Dornfeld, "Automated energy monitoring of machine tools," *CIRP Annals - Manufacturing Technology*, vol. 59, no. 1, pp. 21 – 24, 2010.
[9] L. V. Allen and D. M. Tilbury, "Anomaly detection using model generation for event-based systems without a preexisting formal model," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –15, 2011.
[10] IEEE, "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002), 2008.
[11] D. Buede, *The Engineering Design of Systems: Models and Methods*. John Wiley & Sons, 2009.
[12] A. Vodenčarević, H. K. Büning, O. Niggemann, and A. Maier, "Using behavior models for anomaly detection in hybrid systems," in *Proceedings of the 23rd International Symposium on Information, Communication and Automation Technologies-ICAT 2011*, 2011.
[13] N. Moriz, S. Faltinski, O. Graeser, O. Niggemann, M. Barth, and A. Fay, "Integration und Anwendung von objektorientierten Simulationsmodellen in AutomationML," in *VDI Kongress AUTOMATION 2011*. Baden Baden, Jun 2011.
[14] R. Isermann, "Model-based fault detection and diagnosis - status and applications," in *16th IFAC Symposium on Automatic Control in Aerospace*, St. Petersbug, Russia, 2004.
[15] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *SCIENCE*, vol. 324, April 2009.
[16] L. Console, C. Picardi, and D. Dupre, "Temporal Decision Trees: Model-based Diagnosis of Dynamic Systems On-Board," *Journal of Artificial Intelligence Research*, vol. 19, pp. 469–512, 2003.
[17] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Aug. 2000.
[18] A. Maier, A. Vodenčarević, O. Niggemann, R. Just, and M. Jäger, "Anomaly detection in production plants using timed automata," in *8th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Noordwijkerhout, The Netherlands, Jul 2011.
[19] S. Verwer, "Efficient identification of timed automata: Theory and practice," Ph.D. dissertation, Delft University of Technology, 2010.
[20] A. Vodenčarević, H. Kleine Büning, O. Niggemann, and A. Maier, "Identifying behavior models for process plants," in *16th IEEE International Conference on Emerging Technologies and Factory Automation ETFA'2011*, Toulouse, France, September 2011.
[21] F. Thollard, P. Dupont, and C. de la Higuera, "Probabilistic DFA inference using Kullback-Leibler divergence and minimality," in *Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 975–982.
[22] R. C. Carrasco and J. Oncina, "Learning deterministic regular grammars from stochastic samples in polynomial time," in *RAIRO (Theoretical Informatics and Applications)*, 1999.