First Approaches to Automatically Diagnose and Reconfigure Hybrid Cyber-Physical Systems

Alexander Diedrich¹[0000–0002–8674–6895], Kaja Balzereit¹[0000–0001–9203–5902],
and Oliver Niggemann²

¹ Fraunhofer IOSB-INA, Fraunhofer Center for Machine Learning, Lemgo, Germany
`surname.name@iosb-ina.fraunhofer.de`

² Helmut-Schmidt University, Hamburg, Germany
`oliver.niggemann@hsu-hh.de`

Abstract. Maintaining modern production machinery requires a significant amount of time and money. Still, plants suffer from expensive production stops and downtime due to faults within individual components. Often, plants are too complex and generate too much data to make manual analysis and diagnosis feasible. Instead, faults often occur unnoticed, resulting in a production stop. It is then the task of highly-skilled engineers to recognise and analyse symptoms and devise a diagnosis. Modern algorithms are more effective and help to detect and isolate faults faster and more precise, thus leading to increased plant availability and lower operating costs.

In this paper we attempt to solve some of the described challenges. We describe a concept for an automated framework for hybrid cyber-physical production systems performing two distinct tasks: 1) fault diagnosis and 2) reconfiguration. For diagnosis, the inputs are connection and behaviour models of the components contained within the system and a model describing their causal dependencies. From this information the framework is able to automatically derive a diagnosis provided a set of known symptoms. Taking the output of the diagnosis as a foundation, the reconfiguration part generates a new configuration, which, if applicable, automatically recovers the plant from its faulty state and resumes production. The described concept is based on predicate logic, specifically Satisfiability-Modulo-Theory. The input models are transformed into logical predicates. These predicates are the input to an implementation of Reiter's diagnosis algorithm, which identifies the minimum-cardinality diagnosis. Taking this diagnosis, a reconfiguration algorithm determines a possible, alternative control, if existing. Therefore the current system structure described by the connection and component models is analysed and alternative production plans are searched. If such an alternative plan exists, it is transmitted to the control of the system. Otherwise, an error that the system is not reconfigurable is returned.

Keywords: Model-based Diagnosis · Fault detection and isolation · Reconfiguration.

1 Introduction

Modern production machinery shall act as autonomously as possible [1]. Autonomous machines are characterized by the capability of making and implementing their own decisions regarding resource use and utilization of components. This also includes the capabilities of observing their own behaviour (self-monitoring), diagnosing their faults (self-diagnosis), and restoring valid system behaviour in case of faults (self-healing) [5].

However, there is still no holistic diagnosis and reconfiguration method which can successfully deal with heterogeneous production plant data and the resulting complex models. Most available diagnosis and reconfiguration methods instead tackle sub-problems, such as system modelling, diagnosis logical circuits, or reconfiguration in narrow and controlled domains. Therefore, developing and realising a robust method is still a research challenge.

Performing consistency-based diagnosis is the only known method to realistically find faults in complex systems. Other approaches are heuristic or require every possible system behaviour to be modelled exhaustively. Heuristic models model behaviour against the flow of causality. Based on an input (the effect) they calculate the likely cause of the fault. Models are created in a data-driven way and require a sufficient amount of training examples for each possible output. This amount of training data is often unavailable. Approaches that model every possible system behaviour exhaustively are often limited by real-world constraints such as the unavailability of accurate enough models.

Consistency-based diagnosis brings the advantage that only the normal behaviour of a system needs to be modelled. Thus, no adversarial examples need to be produced (as would be required for heuristic models) and engineers do not need to think of and simulate all possibilities how components can fail. This decreases modelling effort and avoids errors within the model. Additionally it has an advantage over heuristic models as it reasons with the flow of causality using a combination of deduction and abduction. Deduction propagates values from the system input to the system output and shows normal system behaviour. Through abduction deviations from this normal behaviour can be traced back to sets of components which are likely to have caused the fault. Abduction is more similar to the way humans diagnose systems. They analyse the faulty output and then look through the system from the output back to the input until they have identified components whose faulty behaviour might have caused the faulty output.

Figure 1 shows the general concept of the diagnosis and reconfiguration framework. The physical production plant generates process data from its sensors. This process data is discretised in the form of symptoms. A symptom of a signal shows the direction of deviation from normal behaviour (*high*, *low*, or *normal*). Additionally, experts need to provide two kinds of models: a connection model and individual component models. The component models numerically describe the behaviour of each component. For example, a water tank would be modelled by a difference equation. These component models need to be available for each piece of diagnosable equipment within the plant. Further, a system description

Fig. 1. Overview of the automated diagnosis and reconfiguration framework

in the form of a connection model needs to be provided. The connection model is a directed graph showing the causal connections between the employed components. A diagnosis algorithm [15], given the component and connection models and a set of symptoms, computes the smallest amount of possible faulty components that explain the symptoms. To obtain these models it is conceivable to create a digital twin during inception and construction of the plant. The data from the digital twin, such as simulation data, can be extracted and transformed into predicate logic models.

The set of possible faulty components is the input to a reconfiguration algorithm. For reconfiguration, the algorithm takes the structural and component models into consideration [2]. The algorithm does not only search for a new parametrization of the system but also looks for alternative paths that can be used to bypass the faulty components. From these it generates alternative control sequences, which reconfigure plant parameters or use redundant components to keep the plant operating.

We formulate the connection and component models through logical approaches to perform consistency-based diagnosis. Given proper models, we assume the set of symptoms as given. The symptoms can be generated through the use of well-known machine-learning methods such as principal component analysis and artificial neural networks. Diagnosis is realised through an implementation of Reiter's algorithm [15]. The reconfiguration method is based on a combination of causal reasoning and numerical parametrization approaches.

2 State of the Art

Struss [18] published a paper on the fundamentals of MBD of dynamic systems. In this he described how hybrid systems can be modelled without resorting to a complete simulation of the system under investigation. He proposed to capture the temporal and dynamic behaviour of a hybrid system in a set of modes which model the system. Each mode has distinct state and temporal constraints in addition to so called Continuity, Integration, and Derivatives (CID) constraints that affect all modes.

Daigle et al. [4] have adapted a discrete event approach to diagnose continuous systems. They state that each fault that occurs in a continuous system has a unique fault signature. A fault signature denotes a qualitative effect that a fault occurs in an observation. Under the assumption that all fault signatures and measurement orderings are known, they employ a diagnoser that traces the states through a temporal causal graph based on measurements.

Roychoudhury et al. [16] have shown how to use hybrid bond graphs (HBG) to diagnose hybrid systems. HBGs abstractly model the system by describing causal, continuous relationships between components. Daigle et al. [4] have employed the developed HBGs to diagnose a spacecraft power distribution system. Prakash et al. [13] have used an extended framework with HBGs to make improvements in diagnosing two-tank systems.

Grastien [8] used SMT for the diagnosis of hybrid systems. He discretizes values in a hybrid system into a set of distinct states. Each observation $\langle \tau, A \rangle$ is understood as a behaviour A at time τ , where A is a partial assignment of the variables in a state. Each variable is augmented with an indicator stating at which time-step the variable expression is valid.

Fränzle et al. [7] have augmented SMT with probabilistic approaches in order to analyse stochastic hybrid systems. By using bounded-model checking together with probabilistic hybrid automata, piecewise deterministic Markov processes, and stochastic differential equations they are able to create a fault analysis system without the need to formulate intermediate finite-state abstractions as the methods mentioned above do.

In another work, Khorasgani [10] describe a hybrid system model through hybrid minimal structurally overdetermined sets (HMSOs). These are sets of differential equations and (in-) equations which model the behaviour of a hybrid system.

Crow et al. [2] extended Reiter's diagnosis algorithm so it is also capable of determining the components that need to be reconfigured. The components that need reconfiguration are determined in an analogous way as diagnosis is done.

Kobi et al. [11] presented an approach, how to identify and modify the process input. In case of parameter variations, the control input to the system is adapted.

Hwang et al. [9] published a survey on existing fault detection and reconfiguration methods: Most of the existing approaches rely on a quantitative analysis of the system data. Therefore, the numerical values of the system are analyzed. Structural information like a system topology are either not considered or implemented statically into the method.

Fleischanderl et al. [6] and Sabin et al. [17] presented configuration approaches

based on constraint satisfaction. The configuration problem, which is to find an assembly of production tasks given product and production requirements, is mapped onto a constraint satisfaction problem, which task is to find a valid variable assignment subject to some given constraints.

In contrast to Struss [18] and Provan [14] we do not use automata and mode estimation to partition the system into different states. Instead, we only sample the system at some suitable interval and use the obtained information directly to model the states in the state-space representation. Unlike in space-craft, which were analyzed by Daigle [3], fault signatures and measurement orderings are unknown in industrial systems. This requires us to pursue a more uninformed approach. Our approach is an alternative to hybrid bond graphs used by Roychoudhury [16], while they are at the same time an extension to the work of Grastien [8] and Khorasgani [10]. In comparison to Grastien we do not singly use satisfiability modulo theory, but instead capture system behaviour in a state-space representation. We expect this to reduce the required computational effort. We also make use of (in-) equations and differential equations as were used by Khorasgani and Biswas, but augment these with the diagnostic reasoning of traditional model-based diagnosis. Compared to Fränzle, we do not make use of stochastic SMT at this point to keep the system more explainable for users.

3 The multiple-tank model

For this work we will use the four tank system depicted in Figure 2 as a running example. The system consists of four water tanks t , seven electric valves v

Fig. 2. The demonstration use case showing a four-tanks model

with integrated flow sensors, an unlimited water source and an unlimited water sink (not shown). Valve v_0 controls water from the unlimited water source, for example the public water mains, into tank t_0 . From there, three pipes with an equal diameter divide the water flow. Finally, valve v_6 drains tank t_3 into the unlimited water sink, for example a river or a processing facility.

Each tank has two binary sensors which indicate overflow and underflow, respectively. There are no provisions to directly measure the water level. Each valve has a switch which indicates whether or not the valve is open. In addition, each valve has an associated flow sensor.

4 Diagnosing Hybrid Systems

Automatically diagnosing real hybrid systems is a hard task. So far, the only known diagnosis method which can deal with this kind of complexity is consistency-based. The method works by reasoning against the flow of causality, meaning that it uses abduction to determine likely fault causes by evaluating observations. The drawback of this kind of diagnosis is its reliance on accurate models. This diagnosis method requires the availability of three types of inputs. The component models (CM), a connection model (CON), and observations (OBS). In most plants component models are not available and must be obtained through expert knowledge. In the four-tank model and within the process industry in general, these components models are often differential equations or piece-wise functions. The water level in tanks and other tank-like components can be modelled through equations such as

$$\Delta h = \frac{1}{A}(Q_i + C_d a \sqrt{2gh_0}) \quad (1)$$

and discrete switching signals (f.e. from valves) can be modelled with piece-wise functions such as

$$o(h_i, \tau_i^o) = \begin{cases} 0 & \text{if } h_i \leq \tau_i^o \\ 1 & \text{else} \end{cases} \quad (2)$$

The challenge is to obtain these models automatically. Often this can only be done with data-driven models such as max-margin approaches, artificial neural networks, or statistical methods.

The second kind of model is the connection model. CON is a directed graph. The nodes are the individual components whose input and outputs are governed by the component models. The edges of the directed graph show which component input values are related to which component output values. The graph can be obtained by fusing connection information automatically extracted from piping and instrumentation diagrams (P&I) with the component models.

A consistency-based diagnosis algorithm uses the models to calculate the normal system state and merges this information with the observations to perform fault identification. To merge these disparate models they need to be transformed into a suitable logical framework. For this, we have used Satisfiability Modulo

Linear Arithmetic (SMT- $\mathcal{LRA}$). The logic framework allows the formulation of equations, in-equations, and logical predicates as rules and store them in a knowledge base. Within each rule a component is identified through it's health state denoted as $ok(c_i)$. Given CON, an algorithm can determine what effects the failure of a component can have on its surrounding components. Here we focus on describing only normal behaviour. One logical rule for valve 2 in the running example is

$$ok(v_2) \wedge ok(t_0) \wedge ok(t_1) \rightarrow ok(f_2) \quad (3)$$

where v_2 is valve 2, t_0, t_1 are the adjacent tanks, and f_2 is a flow sensor for v_2 . All rules have ok-assumptions for components on the left-hand side and observations on the right-hand side. Reading the rule from left to right uses deduction and tells the algorithm the normal state of the system: "if all components are ok, the flow sensor will show ok readings". For diagnosis the algorithm uses deduction: "Given that the flow f_2 is not ok, the components on the left-hand side are likely candidates". When the rules are created it must be ensured that rules have overlapping sets of components. Otherwise single components cannot be discriminated.

The transformation from the component model into ok-assumptions can be done through standard machine learning algorithms or be integrated into the logical framework itself. For example, the flow through valve 2 can either be calculated using the equations governing the inflow and outflow of tank t_0 , or a simple machine learning algorithm can be trained which outputs ok/nok .

We employ Reiter's diagnosis algorithm [15] to evaluate the generated knowledge base and discriminate faults to obtain a diagnosis that contains the smallest amount of components (minimal cardinality diagnosis).

5 Reconfiguration after faults occurred

After a fault in the system is identified, a reconfiguration method is used to restore valid system behaviour - if possible. So a reconfiguration method works on a sophisticated level it needs to satisfy different requirements: First, the reconfiguration must be done in a short time with a minimal manual effort. Plant downtimes and component failures need to be minimized so that the costs of these errors are reduced. Additionally, the control software needs to adapt to different product specifactions and production modes. Therefore, it must not be static but needs to be able to adapt dynamically. It also needs to handle the complexity of the production plant and therefore consider the system parametrization as well as the system topology simultaneously. A lot of research has been done on the dynamic optimization of the numeric parameters of a system [12, 20, 19]. However, most of these methods only work for a static system configuration and cannot adapt to varying demands. The here presented approach differs from the state of the art since it considers both, the system parametrization and the system topology simultaneously. Thus, complex systems and varying

production demands can be handled by the reconfiguration method. A reconfiguration method also needs to be able to separate between valid and invalid plant behaviour. In general, this information relies on expert knowledge: To separate faulty from non-faulty system behaviour, models representing non-faulty system behaviour are trained based on a set of non-faulty system data. This set has to be determined by an expert. Non-faulty system behaviour may also be invalid, if it does not lead to the required system goal. Thus, an expert has to define the current system goal to make sure that the current configuration leads to this goal.

A reconfiguration method takes a system description consisting of connection model and component models, the current system's state and a definition of the system goal as input. The connection model and the component models are the same as those used for diagnosis. The definition of the system goal is used to determine which system behaviour leads to the correct system goal. For the tank model, a possible system goal is to fill tank t_3 . Every system behaviour leading to t_3 not being filled is invalid. The goal of reconfiguration is to restore valid system behaviour after a fault occurred.

For a reconfiguration the system's properties like its modules and their interconnections are modelled as well as the system goal. The current system state is checked. If it is valid, no action is necessary; the current system behaviour leads to the required production goal. If the current system state is invalid, the necessary system actions to restore valid system behaviour are determined.

Assuming that the pipe connecting t_0 and t_3 is broken and has been identified, based on a diagnosis, as faulty component. The reconfiguration method now returns the control instruction, that the pipe no longer should be used and proposes the connections (v_1, t_1, v_4) and (v_2, t_2, v_5) as alternatives so that tank t_3 is filled.

6 Conclusion and future work

The presented concept considers the automated diagnosis and reconfiguration of hybrid cyber-physical systems. Based on the current system data, a connection model representing the current structure of the system and component models, which model the behaviour of every component, a diagnosis is executed. Thus a set of possible root causes is determined.

After that a reconfiguration method is started: The task of reconfiguration is to restore valid system behaviour after a fault occurred. Given a system goal, the necessary actions for the recovery of the system are determined. Alternative production paths and parametrizations are identified so that the specified system goal still can be reached.

Our future work will be focussed on the automatic extraction of expert knowledge from P&I diagrams and learning components models from data. Currently, the connection model is extracted manually from the system structure. However, this is time consuming and requires a lot of manual effort. To reduce this, the needed information shall be extracted automatically. Creating component mod-

els automatically requires even more research effort. The behaviour of physical components is often governed by differential equations sometimes including nonlinearities. For accurate component models this behaviour needs to be learned and accurately predicted by data-driven models. What models to use and how to train them remains a research challenge.

7 Acknowledgement

This work was developed within the Fraunhofer Cluster of Excellence "Cognitive Internet Technologies".

References

1. Die Bundesregierung. Strategie Künstliche Intelligenz der Bundesregierung. 2018.
2. Judith Crow and John M Rushby. Model-based reconfiguration: Toward an integration with diagnosis. In *AAAI*, pages 836–841, 1991.
3. Matthew Daigle, Xenofon Koutsoukos, and Gautam Biswas. A discrete event approach to diagnosis of continuous systems. In *Proceedings of the 18th International Workshop on Principles of Diagnosis*, pages 259–266, 2007.
4. Matthew J Daigle, Indranil Roychoudhury, Gautam Biswas, Xenofon D Koutsoukos, Ann Patterson-Hine, and Scott Poll. A comprehensive diagnosis methodology for complex hybrid systems: A case study on spacecraft power distribution systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(5):917–931, 2010.
5. European Factories of the Future Research Association. Multi-Annual Roadmap for the Contractual PPP under HORIZONS 2020, 2013.
6. Gerhard Fleischanderl, Gerhard E Friedrich, Alois Haselbock, Herwig Schreiner, and Markus Stumptner. Configuring large systems using generative constraint satisfaction. *IEEE Intelligent Systems and their applications*, 13(4):59–68, 1998.
7. Martin Fränzle, Holger Hermanns, and Tino Teige. Stochastic satisfiability modulo theory: A novel technique for the analysis of probabilistic hybrid systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 172–186. Springer, 2008.
8. Alban Grastien. Diagnosis of hybrid systems by consistency testing. In *24th International Workshop on Principles of Diagnosis (DX-13)*, pages 9–14. Citeseer, 2013.
9. Inseok Hwang, Sungwan Kim, Youdan Kim, and Chze Eng Seah. A survey of fault detection, isolation, and reconfiguration methods. *IEEE transactions on control systems technology*, 18(3):636–653, 2010.
10. Hamed Khorasgani and Gautam Biswas. Structural fault detection and isolation in hybrid systems. *IEEE Transactions on Automation Science and Engineering*, 2017.
11. Abdessamad Kobi, Samuel Nowakowski, and José Ragot. Fault detection-isolation and control reconfiguration. *Mathematics and Computers in Simulation*, 37(2-3):111–117, 1994.
12. Jens Otto, Birgit Vogel-Heuser, and Oliver Niggemann. Automatic parameter estimation for reusable software components of modular and reconfigurable cyber-physical production systems in the domain of discrete manufacturing. *IEEE Transactions on Industrial Informatics*, 14(1):275–282, 2017.

13. Om Prakash and AK Samantaray. Model-based diagnosis and prognosis of hybrid dynamical systems with dynamically updated parameters. In *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, pages 195–232. Springer, 2017.
14. Gregory Provan. Model abstractions for diagnosing hybrid systems. In *Proceedings of the 20th International Workshop on Principles of Diagnosis, DX-09, Stockholm, Sweden*, pages 321–328. Citeseer, 2009.
15. Raymond Reiter. A theory of diagnosis from first principles. *Artificial intelligence*, 32(1):57–95, 1987.
16. Indranil Roychoudhury, Matthew J Daigle, Gautam Biswas, and Xenofon Koutsoukos. Efficient simulation of hybrid systems: A hybrid bond graph approach. *Simulation*, 87(6):467–498, 2011.
17. Daniel Sabin and Eugene C Freuder. Configuration as composite constraint satisfaction. In *Proceedings of the Artificial Intelligence and Manufacturing Research Planning Workshop*, pages 153–161. AAAI Press Palo Alto, CA, 1996.
18. Peter Struss. Fundamentals of model-based diagnosis of dynamic systems. In *IJCAI (1)*, pages 480–485, 1997.
19. Stefan Windmann, Oliver Niggemann, and Heiko Stichweh. Computation of energy efficient driving speeds in conveying systems. *at-Automatisierungstechnik*, 66(4):308–319, 2018.
20. Minjie Zou, Felix Ocker, Edward Huang, Birgit Vogel-Heuser, and Chun-Hung Chen. Design parameter optimization of automated production systems. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pages 359–364. IEEE, 2018.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.