Component-based Combination of Online-Diagnosis Methods Using Diagnostic Directed Acyclic Graphs

Simon Meckel, Roman Obermaisser
Embedded Systems Group, University of Siegen, Germany
simon.meckel@uni-siegen.de, roman.obermaisser@uni-siegen.de

***Abstract*—In safety-critical application domains online fault-diagnosis contributes to a significantly increased system reliability and safety by detecting and diagnosing occurred faults and, if applicable, making the system recover from faults. In order to enable fault-specific recovery actions, e.g., a reconfiguration of the system, cause-based fault identification is needed. This typically requires a large amount of data to be analyzed and evaluated during a diagnostic process. For sound decisions on occurred faults within complex systems, it is often beneficial to combine several online-diagnosis methods. In this work we present a component-based diagnostic framework based on a diagnostic dependency graph. Herein, multiple online-diagnosis methods are combined in the form of encapsulated tasks. The dependencies as well as the tasks adapt to the system at run time. Our diagnostic framework is demonstrated by means of an automotive use-case.**

***Keywords*—Online-Diagnosis; Real-Time; Diagnostic Dependencies; Feature Extraction**

I. Introduction

System level online fault-diagnosis aims at detecting and diagnosing faults in a system at run time by considering a multitude of diagnostic features, derived, for example, from sensor measurements, signal-models, or process models. Since the functionalities of complex electrical or mechatronic systems, amongst others, are achieved through the interactions of multiple components, faults are typically not only reflected in one measurement, i.e., one feature, but rather influence many others. It is therefore no longer effective to observe individual component behavior and straightforwardly map a discrepancy to a root cause, but rather to take all related measurements and features into account, as this allows a more sophisticated root cause analysis. Especially in safety-critical application domains such as avionics, health care or autonomous driving, an acceptable quality of the services of electronic systems is required, even in the event of faults in components or adverse external influences. This requires an online-diagnosis system to take as much knowledge about the system behavior as possible into account, to allow the most suitable fault-recovery possibilities. In this regard, it is often beneficial to combine several online-diagnosis methods instead of utilizing single features from isolated fault-detection methods only. Frequent faults are faults in the design of a component or a system, transient and permanent hardware faults, imprecise specifications or erroneous user operations.

Based on the ever-increasing demand for reliability and safety of industrial systems, amongst others, the last decades yielded a variety of different approaches on fault diagnosis and fault-recovery techniques, which got adapted to different applications, extended or even combined to hybrid approaches. Not only the latest developed industrial machines offer fault diagnosis but the techniques are also integrated into already existing machines and their control systems. The state-of-the-art encompasses a plethora of diagnosis techniques such as signal-based methods, model-based diagnosis, and knowledge-based approaches each with their various realization possibilities. This makes it challenging for engineers to choose or combine the most suitable diagnosis methods for a certain system. All of the aforementioned methods have their distinctive advantages but also different constraints. In order to overcome potential drawbacks and to work out the diagnosis problem in the best possible manner, an integration or combination of two or more diagnosis methods is often aspired in a variety of engineering applications [1]. Due to the different working principles, however, the combination of multiple diagnosis methods is not trivial. Even when reducing the particular methods to their basic working principles, there is no general specification or interface available which offers a straightforward possibility to combine two or more methods. Besides, nowadays implemented diagnosis methods are often static and do not adapt to system or component changes.

In order to fill this gap we develop a component-based framework for diagnosis method combination on a higher abstraction level. The main part is a library of adaptive diagnostic tasks, which are able to process various types of inputs to take account for the working principles of the different diagnosis methods. The overall connection and cooperation (dependencies) of the different diagnostic tasks is organized within a diagnostic directed acyclic graph (DDAG). Herein, the diagnostic process is modeled from a first symptom recognition, i.e., a fault indication, up to the final fault identification, utilizing the benefits of a broad range of diagnostic techniques. Considering that the tasks may be computationally intensive in addition with the fact that the diagnostic process often allows a concurrent processing of different tasks, parallelism can be exploited. In this way, the framework offers an environment for modeling suitable online-diagnosis methods for a variety of technical processes and systems, while adapting to the given infrastructure, e.g., by executing tasks on separate processing units within a distributed network.

II. Online Diagnosis Methods

A great number of state-of-the-art techniques for online diagnosis exists. Model-based diagnosis methods (e.g., [2]) observe the consistency between measured system outputs and model-predicted outputs on the basis of mathematical models. Advanced observer techniques are often utilized for fault identification, i.e., determining detailed fault information such as the type, size, and shape [3]. This diagnosis method requires only a small amount of real time data and is able to even diagnose unknown faults. However, as the needed explicit input-output model of a system is not always available or extremely challenging to obtain, the deployability of model-based diagnosis methods may be limited, potentially restricted to parts of the system. Besides, the diagnostic performance strongly depends on the accuracy of such a mathematical model.

Signal-based fault diagnosis methods extract characteristic features from measured signals, either directly (e.g., limit checking) or through additional data processing (e.g., transforms). Since faults are reflected in the measured signals the features summarize the current health status of the system. For complex systems, multiple features are combined to feature patterns and the diagnostic process evaluates the patterns based on prior knowledge how faults influence the features. In this way, a conclusion about the health status becomes possible at system level, which typically outperforms an individual component behavior observation. Signal-based fault diagnosis is suitable for systems where no model exists or it may be deployed as an extension to systems that are equipped with sensors already but have only been monitored passively. A drawback of the method is that system dynamic inputs such as varying load or power supply fluctuations may degrade the performance, as their impacts may not be represented by the feature patterns [4].

Essential information for fault diagnosis (i.e., the knowledge how faults influence the system behavior and component interdependencies) is also implicitly available in a large amount of historic data of the process, which is often available from previous observations. Knowledge-based diagnosis methods extract this knowledge employing techniques of artificial intelligence such as principal component analysis (PCA), neural networks or fuzzy logic in order to form a knowledge base. By means of consistency checking of the process input and output relations and classifiers the knowledge base is applied to reason about faults. The computational cost and the dependency on a large amount of historical data for training as well as the uncertainty about the diagnostic coverage, i.e., the uncertainty which faults can finally be diagnosed, narrow the fields of application for this diagnosis method.

Depending on the design of the applied algorithm, the depth of fault diagnosis varies from simple fault recognitions up to detailed fault identifications. Due to the fact that all of these fault diagnosis methods have their own advantages and drawbacks, and differently well suit to different problems, hybrid approaches, i.e., combining or integrating more than one diagnosis method arouse recently [1], [5], [6]. The combination, however, is not trivial and often strongly customized for a specific system. Hybrid approaches are often of low generality and low expandability. The majority of the introduced approaches for online-diagnosis with their possibilities of interacting with the system are commonly restricted to the detection of a critical system state, informing an operator to take further actions or performing a transition into a safe system state, e.g., system shut down. A differentiated cause-based reaction according to the identification of faulty subsystems is, if at all, rudimentally employed. The high complexity of state-of-the-art safety-critical systems and the challenge to conclude the correct root cause of a fault at run time necessitate diagnostic algorithms that guarantee correct fault identification within a defined time limit. At best, a general-purpose framework and a library of diagnostic tasks allow different fault diagnosis methods to complement each other in order to gain maximum benefits while keeping the design rules for the diagnostic process as general and simple as possible.

III. Diagnostic Framework

In complex systems, several potential faults (e.g., of different components) may lead to a similar faulty behavior of a monitored signal. In such a scenario, the fault detection is followed by a process of evaluating confirmative and unsupportive diagnostic information, respectively, in order to include or exclude particular faults, eventually specifying the actually occurred fault. The diagnosis of faults requires the execution of different feature extraction, evaluation, and processing tasks in a defined order.

Our component-based approach enables the combination of multiple online diagnosis methods by means of a diagnostic framework including a library of universal and adaptable diagnostic tasks and the associated diagnostic analysis process. The approach aims at utilizing all available knowledge of the process or the system, which potentially comes from different sources (see Section II), according to its most qualified exploitation method, i.e., diagnosis method. The diagnostic graph and tasks are able to adjust at system run time to account for different system states or system changes.

A. Diagnostic Directed Acyclic Graph

We model the dependencies between diagnostic operations in a directed acyclic graph, denoted as $G = (T, E, (\ell_e)_{e \in E})$, with T being a set of tasks t and E being a set of ordered pairs (t, t') modeling a precedence relation between two tasks [7]. Edges from E are also called *(logical) channels*. For a channel e, the number $\ell_e \in \mathbb{N}$ specifies the size of the message that is sent via channel e. A task t' depends directly on t, iff $(t, t') \in E$. A task may depend on multiple others and in turn may work as a prerequisite for other tasks. With the tasks describing the diagnostic techniques and the edges showing the necessary information to be exchanged between tasks, the fault inference process is modeled. When performed at run time on a distributed system the inference on faults is decomposed in the temporal and spatial domain. For complex systems it is

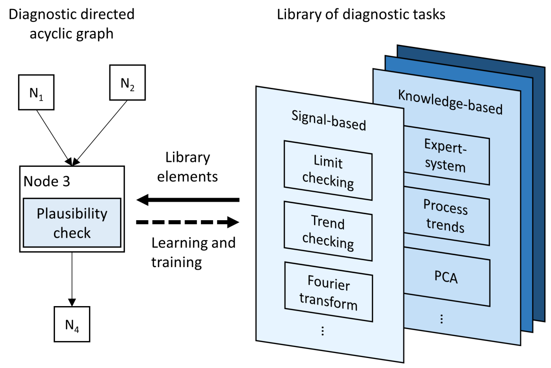

Fig. 1. Example of a diagnostic graph and the library of diagnostic tasks

often not trivial to obtain a DDAG and different strategies exist. In Section IV we introduce an approach based human system expert knowledge and machine learning techniques.

B. Library of Diagnostic Tasks

According to its objective, every task of our diagnostic library utilizes modularized algorithms of the fields preprocessing of sensor or process data, diagnostic feature extraction, and further processing (e.g., combination) of diagnostic information. For example, diagnostic tasks handle the following digital signal and data processing techniques (see also [8], [9]), amongst others:

- Model-based diagnostic tasks
  - Observer based fault classifier
- Signal-based diagnostic tasks
  - Time-domain (limit checking, ...)
  - Frequency-domain (Fourier transform, ...)
  - Time-frequency methods (short-time Fourier transform, wavelet transform, ...)
- Knowledge-based diagnostic tasks
  - Qualitative knowledge-based (expert systems, qualitative trend analysis, ...)
  - Quantitative knowledge-based (principal component analysis, neural networks, ...)

Figure 1 shows a simplified diagnostic graph and an excerpt of some tasks of the library. The node with the ID 3 currently hosts the task *plausibility check*. This task is dependent on data from two another nodes, N_1 and N_2. After processing the result is forwarded to node N_4.

By making the library elements adaptable, the different processing steps can be initially defined according to fixed rules or parameters and can get modified during the run time of the system as more information of the component or the system behavior becomes available. For this, we apply machine learning techniques.

The diagnostic tasks are applied to convert the huge pool of input data to a feature space of essential information, which offers sufficient knowledge about the current health status of the system to distinguish from fault classes through to a full identification of a single specific fault. In this way, a connection between the different diagnosis methods is established. In case of a fault, the attained fault indications initially provide evidence for a certain fault, yet the information may be inconclusive at a certain node of the DDAG (e.g., confirmative information from other nodes is not yet available). The degree of confidence of a correct fault identification increases with more information merged.

C. Adaptable Fault-Inference Process

Due to system resources and computational costs it is not always possible to monitor the entirety of available data for a fault detection within complex systems at all times. One rather concentrates on fewer selected signals and surface symptoms as many potential faults manifest themselves via these signals.

With our component based approach we mainly tackle two difficulties that often limit the performance of diagnosis systems:

The first challenge refers to the fact that diagnostic descriptions of integrated system components are often heterogeneous. For example, the level of detail may vary strongly and depends on the component designer or the supplier. For some components process model equations are available allowing detailed observer-based diagnosis methods. For other components the current status can be determined through sensors measuring physical quantities at various locations within the system or the component. The component behavior also varies depending on external influences or wear, amongst others, such that the diagnostic parameters (e.g., limits for limit checks) must be periodically updated or can be advantageously modified if more knowledge about the components or the entire system is gathered. The component characteristics and corresponding signals can also be extracted (i.e., learned) and refined by observing the component during operation mode. Especially for the automotive industry a huge amount of data can be accumulated with many vehicles in operation.

The second challenge is the fact that component behavior has to be seen in the context of other components. It is often only possible to classify a component's current health status precisely if other measurements, e.g., of neighboring components are taken into account. It is therefore important that the diagnostic dependencies are modeled. Expert knowledge can be used to generate a first diagnostic dependency graph. During the run time of the system or if a modification of the dependencies becomes necessary, e.g., after replacing or adding a component, the DDAG gets adapted and the diagnostic tasks are refined according to component or system changes, merely by evaluating the process data.

IV. Examples and Evaluation

In the following we present an example of a distributed fault-diagnosis process on system level that combines different online diagnosis methods within a DDAG by means of an automotive use-case.

A. Distributed Architecture of Modern Cars

Modern cars have many electronic control units, most of which fulfill specific tasks, e.g., processing sensor data or control signals for other devices. These electronic control units may adopt diagnostic tasks as well, which are related to their predestined task. Thus, the overall electronic architecture of modern cars can be seen as a distributed network where several processing units can execute diagnostic tasks leading to a distributed diagnostic process. In this case the diagnosis system is adapted to a fixed infrastructure (e.g., a network of processing nodes, sensors, ...).

We introduce a Simulink model to exemplarily demonstrate the working principle of a distributed diagnostic process by means of a hybrid-electric vehicle (HEV) model[1]. However, the aspects can be generalized for many other systems or processes.

The HEV-model offers an abstraction of a hybrid-electric car and allows to simulate the car's behavior according to a driving cycle input. The main components are the electrical part with an electric motor, a generator, a voltage converter and a battery. The mechanical part comprises a power split device that combines an internal combustion engine (ICE) with the generator and the driveshaft. Via the power split device the ICE fulfills two tasks: supporting the motor to drive the car and extending the car's operating range by charging the battery via the generator. Besides, the electric motor is directly connected to the driveshaft. A mode logic manages the interaction between these units (e.g., turning on or off the generator depending on the current battery state of charge and driving situation). The model is equipped with a variety of sensors, such as voltage sensors, current sensors, torque sensors or tachometers. The mode logic is monitored as well. Since faults in the system (e.g., failure of a component) are reflected in the measured signals, a diagnostic decision can be on the one hand conducted based on the analysis and processing of fault indications derived from the these signals. On the other hand, discrepancies and change detections of model input-output relations can be applied for components when a detailed analytical model can be assumed.

B. Diagnostic Challenges

In our scenario we find a given system infrastructure (the system, the components, the network, the sensors, ...) and the online diagnosis system is supposed to adapt to this. There are several challenges when a diagnosis system for the HEV-model is to be established. Due to the variety of different sensors and the fact that these signals are influenced by the faults, signal-based fault-diagnosis is predestined, especially the trend and limit checking. However, the importance of certain signals as well as the diagnostic parameters (e.g., the limits) are not obvious. Determining this is often done by human system experts. This may have advantages like experience but may also have disadvantages such as a biased view on the system. In this regard, data-driven feature extraction methods like machine learning help to select the features and set the parameters, while including the information of human system experts by

[1]Hybrid-Electric Vehicle Model in Simulink, http://www.mathworks.com/matlabcentral/fileexchange/28441-hybrid-electric-vehicle-model-in-simulink

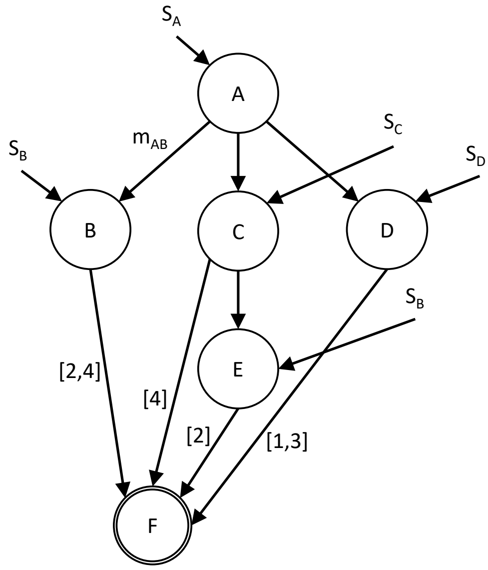

Fig. 2. Diagnostic directed acyclic graph for the example model

defining constraints. Besides, more precise diagnostic results may become possible if components can be characterized by advanced model equations instead of measurements. Models are especially useful to allow a deep diagnosis of the comprising parts of a component. Furthermore, not all components can be diagnosed at all system states with the same diagnostic methods, or diagnostic parameters. In our example, if the generator is currently not in use, it cannot be diagnosed via the signal-based methods applied otherwise. In this case, the dependency graph changes as the generator can be excluded from the list of potential faults. This aspect can of course be generalized.

C. Working Principles of a Distributed Fault-Diagnosis Process

In the following, we demonstrate a fault detection and diagnostic inference process by means of a DDAG. The DDAG in Figure 2 only models the diagnostic dependencies for the faults specified in this example and is established in accordance with the diagnostic model introduced in Section III-A. It is not complete for the whole HEV-model, yet suitable for demonstration purposes. Table I gives an overview of the diagnostic tasks and necessary input signals. In the simplified scenario we assume that only one fault occurs at a time and the potential faults are: failure of the electric motor (label 1), failure of the generator (2), failure of the battery (3), or ICE failure (4).

For the fault inference example the speed controller signal serves as the basis for a surface symptom generation, i.e., multiple faults are manifested herein. A discrepancy of demanded speed and actual speed indicates an unwanted system behavior such as a fault in the system (neglecting external disturbances) or, possibly, a wrong user action. The root cause is yet unknown. Especially in the field of assistance systems for autonomous driving it becomes extremely important to

TABLE I
DIAGNOSTIC TASKS AND SIGNALS

Node	Task (feature gen. and eval.)	Signal	Measurement
A	Motor feature 1	S_A	current
B	Generator feature 1	S_B	current
C	Generator feature 2	S_C	speed
D	Battery feature 1	S_D	current
E	Generator feature 1	S_B	current
F	Final fault decision		

categorize unintended states of the vehicle accurately in order to handle the situation properly.

On a higher abstraction level, we have a catalog of unintended situations that may arise and in contrast we have a catalog of possibilities and actions. For instance, in the described scenario the situation may have arisen from a degrading component, e.g., a problem with the battery. It is then the purpose of the fault-diagnosis system to map this situation to an action, e.g., stop the vehicle immediately, limit certain functionalities to a minimum to prevent greater danger, or tolerate the fault for a certain time.

The DDAG (Figure 2) in its current stage performs fault-diagnosis (feature generation and evaluation) by means of signal-based diagnostic tasks, namely limit checking and trend checking. In the DDAG the labels indicate the faults that are identified at that processing step. The component dependencies are automatically generated and updated via machine learning techniques and additional human expert knowledge (feature processing information). For this, faults are injected into our modified HEV-model and signal variations are observed in order to refine the fault classifications. The order of the diagnostic tasks is arranged according to a *predictor importance* estimate of the machine learning algorithm. Real-world applications may require this method to be modified. The knowledge about fault effects can be gained through fault simulations on real systems, extraction from historic process data, or data fusion from similar processes or systems. As Table I shows, only a small number of measurements is necessary for the fault inference process in our example scenario. In contrast, the potential faults can only be narrowed down to complete component failures which is often not sufficient. If, in a later stage, a detailed battery model can be provided it is included into the diagnostic process to allow a more precise fault identification, including internal battery faults as well. In this case, the DDAG is prepared by means of the machine learning algorithm based on the newly accumulated data.

A stepwise fault-inference process suits real-world systems. Especially when a system to be diagnosed consists of various different components and multiple independent processing units, the relevant data from the applied diagnostic tasks is merged during the fault-diagnosis process via a successive generation, evaluation, and combination of diagnostic features. The ability to adopt the diagnostic methods to a variety of applications may help to increase their reliability, availability, and especially the safety. The framework is particularly advantageous if a system or a process shows strong variations of measurements or model-outputs depending on its current state (e.g., linear vs. scattered value outputs). A continuous modification and adaptation of the diagnosis system can outperform classical, static approaches.

V. Conclusion and Future Work

In this paper we introduce a diagnostic framework consisting of a library of adaptable diagnostic processing tasks in combination with an associated diagnostic analysis process based on a directed acyclic graph. The framework enables to integrate all available knowledge about the system to be diagnosed by combining multiple fault-diagnosis methods. By means of an automotive use-case the method is exemplarily shown. The use-case offers the possibility for extended research and evaluation on the algorithms and the interaction of various diagnosis methods, particularly with regard to temporal predictability and correctness guarantees for fault identification in safety-critical embedded systems with limited resources. Advancing the DDAG to a more powerful meta-model for component-based diagnosis in the future will lead to better support for high-level (semantic) diagnostic task descriptions, handling of diagnostic uncertainties, and evaluation of the diagnostic coverage.

Acknowledgment

This work was supported by the DFG research grants LO748/11-1 and OB384/5-1.

References

[1] D. He, R. Li, and J. Zhu, "Plastic bearing fault diagnosis based on a two-step data mining approach," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3429–3440, 2013.

[2] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[3] K. Zhang, B. Jiang, V. Cocquempot, and H. Zhang, "A framework of robust fault estimation observer design for continuous-time/discrete-time systems," *Optimal Control Applications and Methods*, vol. 34, no. 4, pp. 442–457, 2013.

[4] C. Cecati, "A survey of fault diagnosis and fault-tolerant techniques—part ii: fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Transactions on Industrial Electronics*, 2015.

[5] A. Soualhi, G. Clerc, and H. Razik, "Detection and diagnosis of faults in induction motor using an improved artificial ant clustering technique," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4053–4062, 2013.

[6] N. Sheibat-Othman, N. Laouti, J.-P. Valour, and S. Othman, "Support vector machines combined to observers for fault diagnosis in chemical reactors," *The Canadian Journal of Chemical Engineering*, vol. 92, no. 4, pp. 685–695, 2014.

[7] S. Jo, M. Lohrey, D. Ludwig, S. Meckel, R. Obermaisser, and S. Plasger, "An architecture for online-diagnosis systems supporting compressed communication," in *Digital System Design (DSD), 2017 Euromicro Conference on*. IEEE, 2017, pp. 62–69.

[8] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[9] R. Milne, "Strategies for diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 3, pp. 333–339, 1987.