Quantum Machine Intelligence (2024) 6:66
https://doi.org/10.1007/s42484-024-00184-x

RESEARCH ARTICLE

Solving industrial fault diagnosis problems with quantum computers

Alexander Diedrich[1] · Stefan Windmann[2] · Oliver Niggemann[1]

Received: 21 February 2023 / Accepted: 31 July 2024

Abstract

In this article, we investigate in how far quantum computers can be leveraged to solve NP-complete fault diagnosis problems within the area of industrial cyber-physical systems. Therefore, two approaches are proposed which exploit quantum computing to solve diagnosis problems: The first method employs Grover's algorithm, and the second is based on the Quantum Approximate Optimization Algorithm. To show the industrial application, we present an integrated approach to learn the diagnosis model from process data, check whether the model is suitable, and use it for diagnosis. The result is a method for quantum industrial fault diagnosis. For this approach, the diagnostic capabilities and the runtime have been evaluated on an IBM Falcon processor using three publicly available benchmarks from the process industry. Further, the scaling between quantum computers and classical PCs has been analyzed.

Keywords Fault diagnosis · Quantum computation · Cyber-physical systems

1 Introduction

Industrial cyber-physical systems such as injection molding machines, mixers, bottling plants, and tank systems are usually characterized by high interconnectivity, modularity, throughput, and number of system parameters. Reliable fault diagnosis supports the plant personnel in troubleshooting and contributes to minimizing repair costs and downtime. However, fault diagnosis is a challenging task that demands in many cases exponential runtime (Niggemann et al. 2021). To mitigate the runtime issues, efficient quantum computation for industrial fault diagnosis problems has become an important area of applied research. Nowadays, quantum fault diagnosis faces two problems:

(i) Efficiently solving the diagnosis problem: Many diagnostic procedures are based on reduction to the NP-complete satisfiability problem (Metodi et al. 2014).
(ii) Obtaining available models from heterogeneous cyber-physical systems: In practice, many companies still rely on manual and expert-driven diagnosis and repair procedures.

These, however, are often expensive, slow, and error-prone.

Fault diagnosis itself has been extensively researched in the past. Many different algorithms and methodologies have been developed. But only a few publications are available using quantum computers (all of them use quantum annealers). This article proposes two novel approaches to conduct fault diagnosis of cyber-physical systems on universal quantum computers (QC) and applies them in a general methodology for QC-based industrial fault diagnosis.

Overall, two research questions are answered in this article: How can quantum computers be applied for efficient fault diagnosis in industrial production settings and what are the benefits **(RQ1)**? How is the model for fault diagnosis obtained **(RQ2)**? While the necessity of RQ1 is quite obvious, RQ2 needs some elaboration. In industrial fault diagnosis, each fault diagnosis algorithm requires a suitable model which describes the system's normal behavior (also called a weak-fault model). However, obtaining this model is a difficult task, which depends in many instances heavily

The work of the first author was performed while the author was at Fraunhofer IOSB-INA.

✉ Alexander Diedrich
alexander.diedrich@hsu-hh.de

Stefan Windmann
stefan.windmann@iosb-ina.fraunhofer.de

Oliver Niggemann
oliver.niggemann@hsu-hh.de

1 Helmut Schmidt University, Holstenhofweg 85, Hamburg, Germany

2 Fraunhofer IOSB-INA, Campusallee 1, Lemgo, Germany

Published online: 08 October 2024

on the availability of expert knowledge. To apply a quantum computer to industrial diagnosis tasks, it is thus of utmost importance to also find a practical method for obtaining a suitable model. In practice, experts are necessary to create such a model. An example is an injection molding process, where plastic is heated and pressed through an extruder into a form in which it cools and where the formed plastic is further extracted using robots and placed onto a conveyor belt for further processing, such as painting. The production processes are characterised by high-throughput and low parts cost in order to be competitive. To achieve competitiveness, many production processes use only a minimum of human labor. But as a result, maintenance knowledge (expert knowledge) is only in the hands of a few people who are distributed over a large area on the production floor (Diedrich et al. 2022). Faults can rapidly lead to a standstill and a production bottleneck before they are fixed. It is therefore desirable for companies to automate fault diagnosis.

A diagnosis methodology that is able to create its own models first learns the system components and their interconnections (the diagnosis model) from data. Given such a model, it is then a common approach to use fault diagnosis (De Kleer and Williams 1987; Reiter 1987), where a model is checked for consistency, given current observations (which is NP-complete). If the consistency cannot be established, a diagnosis algorithm determines those components that are responsible for the inconsistency.

While there is some related work on solving the diagnosis problem for combinatorial circuits using quantum computers (Perdomo-Ortiz et al. 2019), there exist so far no diagnosis QC-based approaches for cyber-physical systems. For cyber-physical systems, two challenges exist: (i) The system description SD is hard to find. Usually, SD requires a model formulated in a suitable logic (such as propositional or predicate logic (Grastien 2014) or physical first-principle models (Reiter 1987; Khorasgani et al. 2019). (ii) Cyber-physical systems produce continuous, discrete, and binary data, rendering diagnosis more complicated, as many diagnosis algorithms assume that the set of observations OBS consists of binary data. (iii) Solving a MaxSAT problem is NP-complete (Jung and Sundström 2019).

Let us first look at the basic methodology behind fault diagnosis. Model-based diagnostic approaches are based on assigning the state of health to a set of components, given a model, and a set of observations. This is usually summarized using a tuple (SD, COMPS, OBS), where SD (system description) is the model describing relations satisfied during the normal operation of the cyber-physical system, COMPS (components) is a set of Boolean variables which are true ($\top$), when a component is healthy and false ($\bot$), when a component is faulty. OBS (observations) is a set of observations that indicate the state of the cyber-physical system such as sensor data, setpoints, and quality assurance data. Usually, the system description is formulated within a suitable logical framework (such as propositional logic). In this work, we assume that SD is given in propositional logic, i.e. as AND, OR, and NOT clauses, which are built of the variables from OBS and COMPS. The goal of performing diagnosis is formally to find a satisfying assignment to the expression $SD \wedge COMPS \wedge OBS$, within some logical framework. If the expression is unsatisfiable, a diagnosis algorithm (De Kleer and Williams 1987; Metodi et al. 2014) attempts to find a maximum satisfying assignment (MaxSAT) for the variables in set COMPS, such that the expression becomes satisfiable. The variables which take the value $\bot$ (the minimal unsatisfiable core of the MaxSAT problem), given a consistent model SD and a set OBS, are called the diagnosis ω. Finding maximum satisfying assignment, i.e. a diagnosis with a minimum number of faulty components, can be written as an optimization problem with objective function

$$\mathbf{c}^* = \mathbf{argmax}_{\mathbf{c} \in COMPS^n} \sum_{i=1 \ldots n} c_i, \tag{1}$$

where the components in vector $\mathbf{c} = (c_1 \ldots c_n)$ are treated as binary values ($c_i = 1$ for a healthy component and $c_i = 0$ for a faulty component). The optimization problem is implicitly constrained by the system description SD and the given observations OBS.

In this article, we present a data-driven and explainable fault diagnosis approach and demonstrate its usage on an IBM Falcon quantum processor. For this, we make the following contributions:

- We present the novel algorithm qDiagCPS_Grover that solves the diagnosis problem through the use of Grover's algorithm.
- Following qDiagCPS_Grover, we present the novel algorithm qDiagCPS_QAOA that solves the diagnosis problem through the use of the Quantum Approximate Optimization Algorithm.
- We demonstrate how the propositional logic system description (the model) for both algorithms qDiagCPS_Grover and qDiagCPS_QAOA can be discovered from data using algorithm GTSD. Parts of this algorithm are based on previous work (Diedrich et al. 2024).

The motivation behind the contributions is the following: since diagnosis is an NP-complete problem, it is reasonable to look for more efficient solutions using quantum computers. To realize this, we use two approaches: (a) We convert the propositional logic system description into a SAT problem and use Grover's algorithm (Grover 1997) to find the minimum satisfying assignment. (b) We transform the propositional logic system description into a quadratic binary optimization problem (QUBO), which is solved using the Quantum Approximate Optimization Algorithm (QAOA). We conclude that both solutions find the minimum cardinality diagnosis $\omega' \subset \omega$, with $\omega \in COMPS$ being the set of faulty components. In practice, the presented approach that also learns its system description solves an important auxiliary problem: It enables users of cyber-physical systems to automatically learn models for diagnosis (Yuan et al. 2019). It thus relieves experts and companies of the need to explicitly specify system dependencies and to analytically specify possible faults and their effects.

2 Related work

In the past, related works have shown how combinatorial circuit diagnosis can be performed on adiabatic quantum computers using a QAOA formulation. Most recently, Leipold et al. have presented drivers to perform efficient quantum annealing for combinatorial circuit diagnosis using the well-known ISCAS-85 benchmarks (Leipold and Spedalieri 2022). In another work, Leipold et al. also presented different ansätzes for QAOA in circuit fault diagnosis (Leipold et al. 2022). Bian et al. have presented a method to map constraint optimization problems to quantum annealers through formulating a constraint satisfaction problem (Bian et al. 2016). When using quantum annealers, it is necessary to map each mathematical formulation of a problem onto the special annealing architecture (the layout of the qubits). Doing this mapping through a special constraint formulation can significantly increase the solving efficiency. Finding a suitable mapping for D-Wave quantum annealers was also done by Ding et al. (2024), who performed prime factorization with multiplier circuits. Perdomo-Ortiz et al. have in two publications shown how graph-based systems can be mapped onto quantum annealers and have presented a method on the special case of combinatorial circuit fault diagnosis (Perdomo-Ortiz et al. 2017, 2019, 2015). Mirkarimi et al. have compared how a max 2-sat problem can be solved using quantum (adiabatic quantum computing) and classical algorithms (Mirkarimi et al. 2022). This is similar to the goal and also a motivation of this article, but with the difference that this article focuses on a more general class of MaxSAT in the field of cyber-physical system diagnosis. Feldman et al. have presented the so far only method which performs binary circuit diagnosis using an IBM quantum computer (Feldman et al. 2022). Ajagekar et al. have attempted a rather similar idea to this article, but for the domain of chemical engineering (Ajagekar and You 2022). Fei et al. (2024) have used a similar approach to ours, but applied that thoroughly for power systems. Our approach is more generally applicable for a wider range of cyber-physical systems.

Apart from quantum computing, where the literature is quite sparse, this work is based on causality research for model learning and the research field of consistency-based diagnosis.

The idea of physical causality was introduced by the works of Forbus (1984) and De Kleer and Brown (1984). Recently, Nielsen et al. (2020) have described a causality detection approach for Multilevel Flow Modelling, although compared to our approach they do not deal with fault diagnosis. Jaber et al. (2019) have presented an approach to improve causal reasoning and investigated the use of partial ancestral graphs. Chao et al. (2022) have tried to mitigate the problem of little available data by presenting a framework which uses either physics-based models grounded on first principles, or a convolutional deep neural network approach. Jung and Sundström (2017) have presented a line of research, which makes heavy use of residuals and could be used as an alternative *backend* for the residual generation in this article. Recently, Jung (2022) combine residual generation and neural networks. Our work neither relies on first principle models only nor requires black-box approaches such as neural networks. An active research field between causality research and fault diagnosis is bond-graph approaches. Gao et al. (2015) provide a good overview over bond-graphs and other model-based diagnosis methods. Recently, Borutzky (2020) and Khan et al. (2020) have introduced new residual-based approaches, while Moddemann et al. (2024) and Merkelbach et al. (2024) have introduced alternative methods to learn system descriptions and physical first-principle models.

The notion of fault diagnosis itself in this article is in line with the definitions by De Kleer and Williams (1987) and Reiter (1987). The theory behind fault diagnosis was recently well explained by Feldman et al. (2010), Metodi et al. (2014), and Stern et al. (2014). Our notion of system descriptions was introduced in earlier work by Diedrich and Niggemann (2022) and follows a line of abductive reasoning by Pill and Wotawa (2018) and Bochman (2021). In consistency-based diagnosis, Ignatiev et al. (2019) and Grastien (2014) have presented diagnosis algorithms based on satisfiability modulo theory that are compatible to our approach. Matei et al. (2019) have published articles about diagnosing physical systems using differential equations. Physical system diagnosis has recently been addressed by Muškardin et al. (2020) and by Yucesan et al. (2021). Kolb et al. (2018) has presented a method to learn SMT expressions. But in our approach, there is no information given except the number of symbols and time series data. The theory to check the model validity presented here belongs to the area of diagnosability research. Bittner et al. (2022) have recently published an extensive formalization using error bounds for discrete event systems. All of these approaches, however, are graph-based and focus on analyzing transitions between system states. Another area of research is the definition of cones within circuit diagnosis (Metodi et al. 2014). The formalization presented here is most similar to the idea behind finding cones in Boolean circuit diagnosis. Here, the goal is to extend this idea of cones to the propositional logic expressions of the learned system description SD.

In this article, we mainly look at approaches that combine fault diagnosis and quantum computers. In summary, from an applied standpoint, all of the existing publications have two significant drawbacks: (i) They focus on the well-defined domain of combinatorial circuit diagnosis. By diagnosing these circuits, the architecture as well as the behavior of each component is known beforehand or can be inferred (such as truth tables). This is an important difference to diagnosing cyber-physical systems, which may contain components where no analytical behavior models exist. (ii) All but one of the mentioned prior approaches use adiabatic quantum annealers. These approaches do not exploit the capabilities of general quantum computers, such as the IBM Falcon.

3 Methods

This section introduces three algorithms: qDiagCPS_Grover, qDiagCPS_QAOA, and GTSD. The former two algorithms are alternative approaches to perform consistency-based diagnosis on a quantum computer by using a propositional logic system description and residual values (discretized sensor values that are 0, when no fault exists, and some other real number, otherwise (Diedrich and Niggemann 2022). The goal of qDiagCPS_Grover and qDiagCPS_QAOA is to check whether some diagnosis system tuple (SD, COMPS, OBS) specified in propositional logic is satisfiable. If the tuple is unsatisfiable, the algorithms need to identify the minimal number of components whose faulty behavior may cause the tuple to become unsatisfiable.

The third algorithm GTSD is used to learn a system description from the time series data of a cyber-physical system using a generate-and-test approach. This section will first provide an overview of the developed methodology and then explain the three algorithms in detail.

The general architecture of the presented methodology is shown in Fig. 1. We assume that process data is provided as a time series of the form $((t_0, \mathbf{x}_0), \ldots, (t_{n-1}, \mathbf{x}_{n-1}))$, with $\mathbf{x} \in \mathbb{R}^{n_x}$ denoting the sensor data acquired at time instance $t \in \mathbb{N}$. The process data is used as the input for the algorithm GTSD, which generates a propositional logic system description. Algorithm GTSD uses multivariate Granger causality (Wismüller et al. 2021; Singh and Borrok 2019) to determine the temporal association between the process data and a set of expert-defined component health states. The result is a set of propositional logic expressions, where each expression is of the form $\bigwedge_{c \in COMPS_i} c \rightarrow o_j$, with $COMPS_i \subset COMPS$ and $o_j \in OBS$. Here, $COMPS_i$ is a subset of components, which GTSD identified to be Granger causal to the observations o_j. Each o_j is computed from a residual and is defined as

$$o_j = \mathbf{h}(\mathbf{x}_j, \boldsymbol{\lambda}) = \begin{cases} \top; \text{ no fault} \\ \bot; \text{ else} \end{cases} \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ is a datum of the input process data, and $\boldsymbol{\lambda} \in \mathbb{R}^{n_\lambda}$ is a set of parameters. $\mathbf{h}(\cdot) : \mathbb{R}^{n_x \times n_\lambda} \rightarrow \mathbb{B}$ is usually implemented through thresholds, analytical equations, or through statistical approaches (Gao et al. 2015; Jung and Sundström 2019).

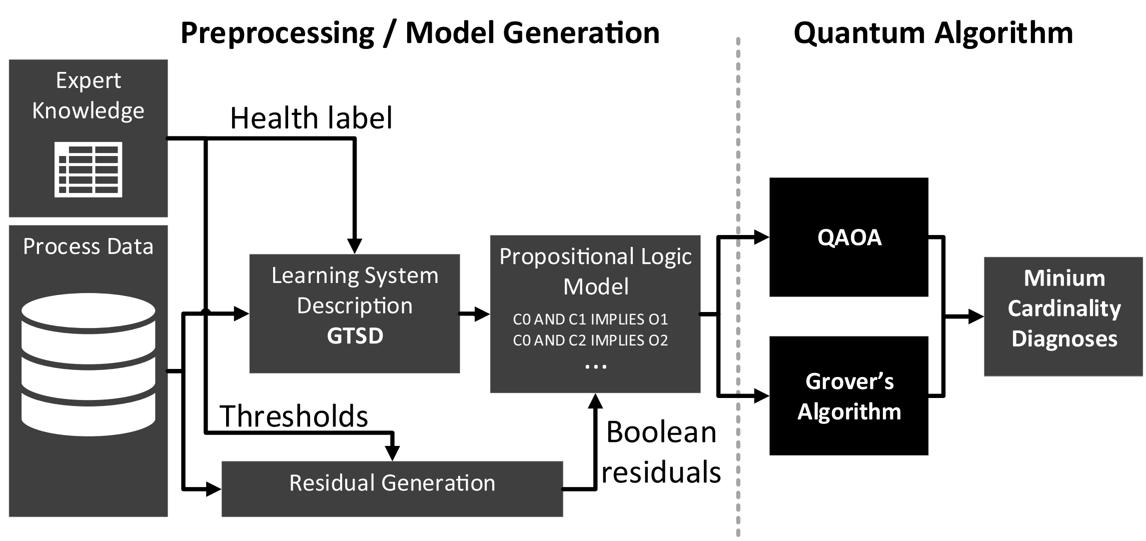


Fig. 1 General architecture of the proposed methodology. The contribution is how the well-known quantum algorithms QAOA and Grover can be applied to learned system description to perform fault diagnosis of industrial cyber-physical systems

For quantum computation, we use two different formulations: One is a quadratic unconstrained binary optimization (QUBO) formulation solved by QAOA (Quantum Approximate Optimization Algorithm). For this, the propositional logic formulation (SD, COMP, OBS) is converted into an optimization problem. The goal is to find a minimum set of components, which satisfies the constraints resulting from the system description and the observations. Therefore, SD and OBS are automatically transformed into a set of linear equations. The second approach is using Grover's algorithm. For this, a phase oracle is created from SD and expert knowledge. Then, the phase oracle and SD are used as input to the Grover amplification problem. The result is a health assignment to all of the components within SD, which corresponds to the set of the minimum cardinality diagnoses.

We will now first describe the quantum algorithms by assuming a given system description and a set of observations. Then, we will present the generate-and-test approach to obtain system description from process data.

3.1 Using Grover's algorithm

According to Metodi et al. (2014), the diagnosis problem can be translated into a SAT problem. As the translation from classical computers to quantum computers is still limited to a narrow set of methods, we chose to encode the diagnosis problem as a SAT problem and then apply Grover's algorithm to solve the problem on a quantum device. Grover's algorithm (Grover 1997) was published in 1997 and enables practitioners to solve combinatorial problems with the help of a predefined oracle. To diagnose cyber-physical systems, the algorithm is used to solve a MaxSAT problem. Given a number of propositional logic clauses (the system description) $\gamma_0 \ldots \gamma_{n-1}$, $n \in \mathbb{N}$, the task for a solver is to find an assignment to the variables in each clause γ, such that the maximum number of clauses is satisfied (i.e., evaluates to $\top$). For this, Grover's algorithm starts with a set of Boolean variables $\{c_0 \ldots c_{n-1}\}$, where in the case of cyber-physical system, diagnosis variables c are the health states of COMPS, i.e., $c \in COMPS$. The algorithm also requires a known fault annotation, which we denote as $COMPS^H$. This represents a complete health assignment to each variable in $COMPS$ for each fault state. Please note that for the experiments in this article, we only used single fault examples. Therefore, $COMPS^H$ is a one-hot encoding, where all variables are assigned $\top$ except one. Grover's algorithm starts with a uniform probability distribution over all variable COMPS. In quantum terms, a Hadamard transformation H is applied to initialize the state of the n qubits of the quantum system:

$$|c_0 \ldots c_{n-1}\rangle = H^{\otimes}|0 \ldots 0\rangle \tag{3}$$

Then, an oracle is employed to evolve the state into a target state. The oracle function is treated as a black-box which outputs $\top$, when $COMPS^H$ is hit and $\bot$ otherwise. More formally, an oracle $\mathcal{O}$ performs the function $\mathcal{O}|COMPS\rangle = (-1)^{f(COMPS)}|COMPS\rangle$, where $f(COMPS)$ is a function that outputs $\top$, if $COMPS^H$ was hit.

The method to adapt Grover's algorithm for cyber-physical systems is listed in Algorithm 1. qDiagCPS_Grover takes as input set $(SD, COMPS, OBS)$. The algorithm first reads process data from a cyber-physical system. It then translates the expressions of SD into propositional logic expressions in conjunctive normal form (CNF). Using the propositional expressions, an oracle is created and Grover's algorithm is called. The result is the minimum cardinality diagnosis ω'. We will now explain each of the functions in detail. Please note that some of the functions are also used in qDiagCPS_QAOA below.

Algorithm 1 Quantum diagnosis for cyber-physical systems using Grover's algorithm.

```
procedure qDiagCPS_Grover((SD, COMPS), T)
    M ← getReadings()
    OBS ← generateSymptoms(M, SD)
    F_cnf ← toCNF((SD, COMPS, OBS))
    oracle ← generateOracle(F_cnf, T)
    ω' ← Grover(F_cnf, oracle)
    return(ω')
end procedure
```

The function $getReadings() : \mathbb{R}^n \rightarrow \mathbb{R}^n$, with $n \in \mathbb{N}$, obtains time series data from some cyber-physical system hardware. Through abuse of notation, we say that the input to the function is $\mathbb{R}^n$, although the input is read from the actual cyber-physical system, instead of being a parameter. Time series data is a set of tuples $(t, s_0, s_1, \ldots, s_{n-1})$, where $s \in \mathbb{R}$ is some real value and $t \in \mathbb{N}$ is some timestamp. If needed, the function may implement noise processing, scaling, centering, NA-value removal, or other common pre-processing techniques.

Function $generateSymptoms() : \mathbb{R}^n \rightarrow \{\top, \bot\}^*$ with $n \in \mathbb{N}$ uses the obtained sensor readings, generates residuals using $h()$ Eq. (2), and applies some discretization function from the piece-wise $h()$ to $\{\top, \bot\}$ for each residual to obtain Boolean observations. Formally, function $h()$ takes a snapshot of the current system's state, input, and output. Sensor readings are categorized into state variables (such as water levels in tanks or a capacitor's charge), input variables (inflow, input voltage), and output variables (outflow, output current). Then, individual residuals are generated by assigning one or more of those categorized sensor values to multiple health functions $h()$. In practice, $h()$ can also be evaluated for single signals. The discretization is usually done through thresholds. This would result in, for example, some expression $m \leq \lambda_0$, for some threshold $\lambda_0 \in \mathbb{R}$, and some sensor value $m \in \mathbb{R}$.

Function $toCNF() : (SD, COMPS, OBS) \rightarrow F_{cnf}$ converts the propositional logic tuple $(SD, COMPS, OBS)$ into a conjunctive normal form for further processing. Given the system description and the components, the function first integrates the residuals. This is done by assigning each variable o_j within the propositional logic expressions $\phi_i : \bigwedge_{c \in COMPS_i} c \rightarrow o_j$ some discretized sensor value through Eq. (2).

The function $generateOracle() : F_{cnf}, \mathcal{T} \rightarrow \{\top, \bot\}*$ provides a one-hot encoding for the target state, given the current system model in conjunctive normal form and the ground truth $\mathcal{T}$, with $\mathcal{T} \in \{\top, \bot\}^*$. In practice, the oracle is a callback function, which is used by the Grover algorithm to determine the goal state for the fault diagnosis.

Performing the MaxSAT calculation for diagnosis is done through function $Grover() : F_{cnf}, oracle \rightarrow \omega$, which takes the conjunctive normal form and the oracle as its input and computes the minimum cardinality diagnosis.

3.2 Quantum approximate optimization algorithm (QAOA)

Alternatively to the Grover algorithm, the diagnostic problem is formulated as a quadratic unconstrained binary optimization (QUBO) problem that can be efficiently solved on a quantum computer using the QAOA algorithm. Starting from the problem description in Section 1, the linear cost function in Eq. 1 is adopted directly, while the system description SD and the observations OBS introduced have to be converted into a linear constraint set **CS**, leading to the QUBO

$$\mathbf{c}^* = \text{argmax}_{\mathbf{c} \in COMPS^n} \sum_{i=1 \ldots n} c_i \quad \text{with resp. } \mathbf{CS} \tag{4}$$

To obtain the linear constraint set **CS**, the replacements shown in Table 1 are successively carried out on the system description SD.

In this process, the AND, OR, and NOT sub-clauses in the clauses of the form $\gamma \rightarrow \bot$ (rules 1–3) are simplified until they are reduced to single variables $\gamma = z$, so that substitution rule 4 can be applied:

- Propositional logic clauses of the form $\gamma(x \wedge y)$ are each replaced by a simplified clause of the form $\gamma(z)$ and the additional condition $z = x \wedge y$. The condition $z = x \wedge y$ is replaced according to Bisshop (2019) by three linear inequalities, which are used as constraints of the quadratic optimization problem.
- Propositional logic clauses of the form $\gamma(x \vee y)$ are each replaced by a simplified clause of the form $\gamma(z)$, with $z = x \vee y$, where the condition $z = x \vee y$ is replaced by three linear inequalities (see also Bisshop (2019)).
- Propositional logic clauses of the form $\gamma(\neg x)$ are each replaced by a simplified clause of the form $\gamma(z)$, with $z = \neg x$, where the condition $z = \neg x$ is written as linear equation $z = 1 - x$.

Expressions of the form $\gamma \rightarrow \top$ do not provide any information about the faulty components due to the equivalence

$$\gamma \rightarrow \top \leftrightarrow \neg\gamma \vee \top \leftrightarrow \top \tag{5}$$

and can therefore be removed from the constraint set (rule 5 in the replacement rule table).

The solution of the resulting QUBO is carried out using the QAOA algorithm.

Algorithm 2 Quantum diagnosis for cyber-physical systems using QAOA.

```
1: procedure QDIAGCPS_QAOA(SD, COMPS)
2:     M ← getReadings()
3:     OBS ← generateSymptoms(M, SD)
4:     F_cnf ← toCNF((SD, COMPS, OBS))
5:     qubo ← CNFtoQUBO(F)
6:     ω' ← qaoa(qubo)
7:     return(ω')
8: end procedure
```

Algorithm 2 shows the steps necessary to perform QAOA optimization. After the pre-processing in steps 1–4 of Algorithm 2, a QUBO problem is generated by repeated application of the proposed transformation rules, which are given in Table 1 (step 5). Then, the QUBO is solved on the

Table 1 Replacement rules to transform the system description into a set of linear inequalities

Rule	Original constraint set	Replaced constraint set
1	$\{\gamma(x \wedge y) \rightarrow \bot\}$	$\{\gamma(z) \rightarrow \bot, x - z \geq 0, y - z \geq 0, x + y - z \leq 1\}$
2	$\{\gamma(x \vee y) \rightarrow \bot\}$	$\{\gamma(z) \rightarrow \bot, z - x \geq 0, z - y \geq 0, z \leq x + y\}$
3	$\{\gamma(\neg x) \rightarrow \bot\}$	$\{\gamma(z) \rightarrow \bot, z = 1 - x\}$
4	$\{z \rightarrow \bot\}$	$\{z = 0\}$
5	$\{z \rightarrow \top\}$	$\emptyset$

quantum computer using QAOA. In the end, the minimum cardinality diagnosis is returned.

3.3 Learning a system description from data

In practice, a common problem when diagnosing faults in cyber-physical systems is the unavailability of suitable system descriptions. If the system description is not available, incomplete, or is not expressive enough, faults may not be diagnosed correctly. Since the focus of this article is on practical methods to perform fault diagnosis of cyber-physical systems using quantum computers, it is necessary to also provide a method to obtain suitable system descriptions.

The intuition behind the algorithm introduced in this section is that algorithms exist that learn dependencies between signals in time series data. Since time series data is also usually readily available in cyber-physical systems, we propose to employ such an algorithm to identify dependencies between health annotations and process signals. The result is a propositional logic system description that we improve iteratively and which can then be used as input to the algorithms qDiagCPS_Grover and qDiagCPS_QAOA introduced above.

This paragraph explains how to automatically obtain a system description (SD) in propositional logic from process data (Sugihara et al. 2012) and test its usefulness. Here, we will first describe the theory behind model learning and testing in the context of diagnosis. Then, we will introduce the generate-and-test algorithm GTSD to obtain the system description automatically. The main goal of the algorithm is to obtain a useful system description SD. To obtain a single logical expression, an algorithm needs to find those components within the set of all components COMPS, whose change in health status corresponds to a change within the set of residuals OBS. In other words, the algorithm must find all those components, whose faulty behavior is recognized by a residual.

Granger causality is an often applied method to find Granger causal dependencies in time series data, which is the primary type of data available in cyber-physical systems (Singh and Borrok 2019; Wismüller et al. 2021). Using Granger causality, it is possible to compute whether one signal has a significant effect on some other signal over time. Granger causality is defined as follows: Given two variables $X \in \mathbb{R}$ and $Y \in \mathbb{R}$, predicting future values of X is compared to predicting future values of X and Y. If the prediction is better, when predicted through X and Y, it is said that Y Granger causes X. In practical terms, this means that when some time series of a temperature can be extrapolated better using the temperature and pressure values, compared to the temperature values alone, then the pressure Granger causes the temperature. The main benefit of Granger causality compared to, for example, definitions from Pearl (2009) is that it can be learned from data and has a lower computational cost, which makes it suitable for the use with cyber-physical systems.

To use Granger causality, a matrix with a fault injection is created of the form

$$F = (c_0, c_1, c_2, ..., c_{n-1}), \tag{6}$$

where $c \in F \subset COMPS$. Each c_i is a named Boolean variable indicating if the component is faulty at a given time. Table 2 shows exemplary data. At time points $i \in \mathbb{Z}$, a fault is injected, where component c_0 is set to faulty. Columns r_0 to r_{q-1} are residuals generated through (2) (Diedrich et al. 2019). Applying Granger causality to the elements of Table 2 yields a set of tuples with variable length of the form $\{(c_0, c_1, ...c_i, o_j), \dots\}$, where o_j is a residual within the set of residuals $\mathcal{R}$ and $c_j \in COMPS$.

The generated set of tuples is converted into a propositional logic system description by *inserting* conjunctions between the component and an implication symbol from the components to the residual. After having constructed SD from process data, it is necessary to check whether most (or even all) faults can be distinguished.

It must be noted that algorithm GTSD by design is neither sound nor complete. Relying on data-driven methods such as Granger causality makes it impossible to guarantee that all physical dependencies will be found (completeness) and that only valid physical dependencies are identified (soundness). To mitigate this weakness, we suggest using the proposed generate-and-test approach, in which we generate a system description, check if all the components can be diagnosed, and then adjust the parameters of the identification procedure accordingly. How we determine whether a component is diagnosable is the subject of the next subsection.

3.4 Checking diagnosability of a system description

Without an automated diagnosability check, it requires experts to thoroughly check the generated system description. For example, a system description might be generated, where faults are masked, which would prohibit diagnosis algorithms to find the right root cause. Here, we only use the set of component health states $F \in \phi$ (which means, we dis-

Table 2 Expert defined components $c \in F$ related to residuals $r \in R$. $i, k, q \in \mathbb{Z}$

t	c_0	c_1	$\cdots$	c_{n-1}	r_0	r_1	$\cdots$	r_{q-1}
0	$\bot$	$\bot$		$\bot$	$\bot$	$\bot$		$\bot$
i	$\top$	$\bot$		$\bot$	$\bot$	$\top$		$\bot$
$i+1$	$\top$	$\bot$		$\bot$	$\bot$	$\top$		$\bot$
k	$\bot$	$\bot$		$\bot$	$\bot$	$\bot$		$\bot$

card the residual, which is only used for diagnosis). Checking diagnosability is done through the single fault assumption, meaning that each single fault should be diagnosable through the observations. The components $c_i \in F_i \subset F$ are those components, which were associated with exactly one residual o_j.

Definition 1 (Diagnosability of single faults) Given some sets $F_i, F_j \dots \subset F$ of component health states with different components, a component is diagnosable, if there exists a single component c_i with $c_i \in F_i \wedge c_i \in F_j \wedge \dots$ for any $i, j \in \mathbb{Z}$. A component is always diagnosable, if it is the single component associated with a unique residual.

Diagnosability for a component is defined such that the component is not *masked* (or within a cone, when using the terminology of Metodi et al. (2014)). This is calculated by checking whether the component health can be determined through multiple tuples (multiple expressions) within F. At the same time, it must be ensured that no other component is solely observable through the exact same rules. Applying the diagnosability check for all components over the whole set F makes it possible to determine perfect diagnosability.

Definition 2 (Perfect diagnosability) Perfect diagnosability exists, when $\forall c_i \in F$ at least two subsets exist, such that $\exists F_i, F_j : c_i \in F_i \wedge c_i \in F_j$ and c_i is the only component in those two sets, with $i, j \in \mathbb{Z}$ and $F_i, F_j \subset F$.

If each component is observable through at least two sets F_i, F_j and no other component is observable through the same observations, the set exhibits perfect diagnosability. In practice, this is hard to achieve, especially when the system description is learned from data. For those components which cannot be perfectly observed, the term hidden components is defined.

Definition 3 (Hidden components) Hidden components are non-diagnosable components, where for at least two components $c_i, c_j \in F_m$ and $c_i, c_j \in F_l$, with $F_m, F_l \subset F$ and $i, j \in \mathbb{Z}$.

These are sets of components where, given the learned system description, no observation can determine which of the components is faulty. Instead, a diagnosis algorithm must assume that once all other components are exonerated, the fault must be located within the set of hidden components. This assumption, however, increases the achievable minimum cardinality diagnosis. In Algorithm 3, lines 14 to 25 perform the diagnosability check and output $\emptyset$, when perfect diagnosability exists and output the set of hidden components F', when components cannot be distinguished from each other given the system description.

$$\omega_q = \frac{\sum_{f_i' \in F'} |f_i'|}{\sum_{f_i \in F} |f_i|} \tag{7}$$

Given the theory about Granger causality and checking for diagnosability, it is now possible to formulate an algorithm which integrates both methods to iteratively create system descriptions.

3.5 Algorithm to learn system descriptions iteratively

Algorithm 3 creates propositional logic diagnosis rules from process data through a generate-and-test approach. We will first outline the general approach and then discuss the individual methods in detail. Input to the algorithm is a time series T and thresholds λ. Lines 2 and 3 in Algorithm 3 preprocess T, such that only signals with legal values and variance are contained. Then, F is merged with the residuals $\mathcal{R}$ in line 4, to create a matrix with component health states and residuals. Next, we use the Akaike information criterion (AIC) (Akaike 1998) to determine the optimum lag for the Granger causality test (a necessary parameter). The lag determines how far into the past of a sensor value the Granger causality test looks. The multivariate Granger causality test itself returns a matrix with p-values ρ, which are sorted such that only those dependencies are selected, which are likely to cause changes in some component c_i (according to threshold λ_p). From those selected dependencies, tuples are created that associate a residual to a set of one or more dependent components. Then, those tuples are converted into propositional logic expressions of the form $\bigwedge_{F_i} c_j \rightarrow o_l$, with $F_i \subset COMPS$ and $i, j, l \in \mathbb{N}$.

Using the propositional logic statements ϕ, the algorithm checks iteratively whether components are diagnosable. The set is sorted according to cardinality with ascending order, such that rules with the least number of components are visited first. Then, for each component, it is checked in how many other rules the component is represented. If the component is the only member in its set, it can be diagnosed (this is the trivial case). Components that can be distinguished in two or more sets are set as diagnosable. By dividing the number of hidden components within the output set F' through the number of components within the input set F, it is possible to calculate a quotient. The quotient can be used to obtain a quantitative measurement about the expressiveness of a learned system description. In other words, it determines how many components can be successfully diagnosed.

Generating a system description and testing it is executed as long as there are undiagnosable components or until a specified timeout is reached. During this, the optimal lag is adjusted through a random walk (line 27).

The individual functions are now explained in detail.

Function extractComps(): $T \rightarrow F$ with T being a time series of component health states analyzes which components are relevant at the time of ϵ. In general, the function

Algorithm 3 GTSD: generate and test approach for learning system descriptions.

1: **procedure** GTSD(T, λ)
2: $F \leftarrow extractComps(T)$
3: $\mathcal{R} \leftarrow makeResiduals(T, \lambda)$
4: $\xi = F \circ \mathcal{R}$
5: $optiLag \leftarrow aic(\xi)$
6: $F' \leftarrow F$
7: **while** $F' \neq \emptyset | \neg timeout$ **do**
8: $\rho \leftarrow granger(\xi, optiLag)$
9: $\phi \leftarrow \emptyset$
10: **for** $c_i, elem \in \rho$ **do**
11: **if** $elem > \lambda_p \in \lambda$ **then**
12: $\phi \leftarrow \phi \cup elem$;
13: **end if**
14: **end for**
15: $F' \leftarrow F$
16: $F_{sorted} \leftarrow \phi.sortBy(smallest)$
17: **for** $card = [1..max(|F_{sorted}|])$ **do**
18: $F_c \leftarrow F_{sorted}.setsWithCardinality(card)$
19: **for** $f_i \in F_c$ **do**
20: **for** $c_{f_i} \in f_i$ **do**
21: **if** $\exists c$, where $c \in (f \setminus \{f_i\}) \land c \neq c_{o_j}$ **then**
22: $F' \leftarrow F' \setminus c_{f_i}, \forall f_i \in F'$
23: **end if**
24: **end for**
25: **end for**
26: **end for**
27: $optiLag \leftarrow optiLag + 1$
28: **end while**
29: $return(F', \phi)$
30: **end procedure**

puts exactly those data points into F that belong to components (c_0, c_1, etc.). Please note that input to the algorithm is for practical reasons a single time series T. As a result, the component health states within the time series need to be extracted.

Function makeResiduals(): $T, \lambda \rightarrow F$ with T being a time series of residuals and λ being a set of parameters (threshold values). Similar to function extractComps(), the function puts exactly those data points into F that belong to residuals (o_0, o_1, etc.). Since T usually contains real-valued residuals, the function internally needs to discretize these into Boolean ones.

Function aic(): $\xi \rightarrow \mathbb{N}$, with ξ being a tuple of component health states and residuals, computes the Akaike information criterion (Akaike 1998) for different models. The output is an estimation for the lag parameter, which is used for Granger's algorithm. The lag determines how far into the past Granger's algorithm looks. For example, if the Akaike information criterion determines the lag to be 2, then Granger's algorithm will compute causality by analyzing the effect of the last two values of one signal on some other signal. If the lag is 10, it would analyze the effect of the last ten values.

The function granger(): $\xi, optiLag \rightarrow P$, with ξ being a tuple of component health states and residuals, $optilag$ being the lag determined using the Akaike information criterion, and P being a set of tuples. The output of the function is a set of tuples, whose rightmost element denotes the residual and the elements of the left of it denote the dependent (Granger causal) components $(c_i, c_j, \ldots, r_k)$, with $i, j, k \in \mathbb{N}$.

Lines 15 to 26 take as input the set Φ and check iteratively, whether components are diagnosable. The set is sorted according to cardinality with ascending order, such that rules with the least number of components are visited first. Then, for each component, it is checked how many other rules the component is represented in. If the component is the only member in its set, it can be diagnosed (this is the trivial case). Components that can be distinguished in two or more sets are set as diagnosable.

4 Results

Empirical results

The proposed methodology was implemented using the IBM Qiskit software and run on a 27-qubit IBM Falcon processor for the quantum experiments and on a 64-bit Intel Core i7-9750 H CPU and Windows 10 operating system with 16GB RAM for the pre-processing and model generation. The diagnostic approach was evaluated in four types of systems: (i) a benchmark published by Balzereit et al. (2021) containing different examples of multiple tank-systems. (ii) A benchmark modelling a bottling plant presented by Ehrhardt et al. (2022). These are referred to as follows: DS1 is a mixing system, DS2 is a distill, DS3 is a filter, and DS4 is a bottling system. (iii) A benchmark of the well-known Tennessee Eastman Process (abbreviated as TE) by Brown (1974). (iv) Four toy examples *Example 1*, *Example 2*, and two synthetic examples derived from the benchmarks. To increase system size, we added the system description of the smaller systems without duplicating statements and thus obtained the synthetic examples *DS1+TE* and *DS1+DS4+TE*. In these examples, we added the system descriptions of several systems to enlarge the system model. For example, for *DS1+TE*, we added the system descriptions of *DS1* and *TE* without duplications. All of the mentioned examples can be compared to use cases in the process industry.

As an example, the propositional logic system description of the bottling process of the benchmark presented by Ehrhardt et al. (2022) is considered in detail. The system consists of two tanks, a pump, and two valves. Residuals o_0 to o_4 are assumed to take on the value of $\perp$. The result of the GTSD algorithm is a set of expressions, in which only those expressions, where the right-hand side has adopted the value $\perp$, are relevant for the diagnosis. This reduced set of expressions is used as the input for the proposed diagnostic approaches.

$$\begin{aligned} pumpP401 \wedge tankB402 &\rightarrow o_0 \\ tankB401 \wedge pumpP401 \wedge tankB402 \wedge valveDiscrete \wedge valveV403 &\rightarrow o_1 \\ tankB401 \wedge pumpP401 \wedge tankB402 &\rightarrow o_2 \\ pumpP401 \wedge tankB402 &\rightarrow o_3 \\ tankB401 \wedge pumpP401 \wedge tankB402 \wedge valveDiscrete \wedge valveV403 &\rightarrow o_4 \end{aligned} \tag{8}$$

The goal of the diagnostic approaches is to find the minimum hitting set for these expressions. In the Grover algorithm, the additional assumption is made that only a single fault occurs in the system at a given time, while the QAOA approach is capable to find multiple simultaneous faults. In the case of the example, the minimum hitting sets are the minimal cardinality diagnoses $\omega' = \{\{tankB402\}, \{pumpP401\}\}$.

Table 3 provides information about the parameters of the investigated systems. The names of the systems are given in the leftmost column. Column |COMPS| denotes the size of the set of components and shows how many different components are contained in the system description. Column |OBS| denotes the size of the set of faulty observations (the overall number of observations is higher).

The evaluation results for the different approaches are given in Table 4. Grover (Q), Grover (C), QAOA(Q), and QAOA (C) denote the implementations of the Grover approach and the QAOA approach on a quantum computer and on a classical computer, respectively. For each approach, the size of the result set is given, which is a permutation of the component health states. A bold number indicates that the obtained result set is identical with the actual set of injected faults. The actual size of the set of injected faults is given in the last column. QAOA and Grover have different mappings to the qubits on the quantum processor (Perdomo-Ortiz et al. 2019).

Table 3 Experimental results

System	|OBS| faulty	|COMPS|
Example 1	2	3
Example 2	2	4
DS1	4	7
DS2	2	3
DS3	4	6
DS4	5	5
One tank	4	3
TE	5	4
Three tanks	6	4
DS1+DS4+TE	6	12
DS1+TE	5	11

|OBS| is the number of observations (residuals) that were faulty, and |COMPS| is the number of components within the system

Therefore, only systems up to a size of five components could be solved with the QAOA approach. Compared to Grover's algorithm, the diagnostic results are significantly better (in most cases, the correct diagnosis is obtained) and the size of the minimum cardinality diagnosis is significantly smaller, but the execution time is much longer. The results are shown in Table 5. By design, when using Grover's algorithm, a ranked list of solutions is computed, where the goal is to have the correct solutions at the top of the list. When looking at the actual results, we see that the correct solutions are among the produced solutions, but are not necessarily at the top of the result list. Therefore, either the oracle used to guide the solutions for Grover's algorithm needs to be tuned or post-processing needs to be executed to further discriminate candidate solutions. However, using a supervised approach or through integrating partial expert knowledge, we believe that qDiagCPS_Grover is a faster alternative for qDiagCPS_QAOA and can thus also generate useful results.

For the displayed runtimes, it must be noted that executing the quantum experiments is done by uploading the Python Qiskit code onto an IBM Quantum Runtime, which then executes the code batch-wise. As such, the time measurement always contains a certain amount of classical computation. What can be concluded from the runtime analysis then is only the qualitative behavior of the systems under investigation.

This can also be seen in Fig. 2a and b for the Grover algorithm. While the absolute numbers are higher (partially due to the server computation) in the quantum case, the scaling seems to be slightly better.

The same is true in Fig. 3a and b for the QAOA algorithm. Due to differences in computation, the absolute numbers differ from those of the Grover algorithm, and the results exhibit a higher standard deviation. However, the differences in scaling become evident when looking at Table 6.

A comparison shows that despite the exponential behavior of the quantum algorithms, the exponential behavior of the classical computer is worse (see Table 5). To prove this empirically, we have determined the quotient between the mean of the experiments with the smallest number of components (three components for all systems) to the mean of the experiments with the largest number of components (6 and 12, respectively) and summarized the findings in Table 6. For both algorithms, QAOA and Grover, it is evident that

Table 4 Experimental results

System	Grover (Q)	Grover (C)	QAOA (Q)	QAOA (C)	Exp
Example 1	8	8	**1**	**1**	1
Example 2	16	16	**4**	3	4
DS1	120	120	–	–	4
DS2	13	**2**	**2**	**2**	2
DS3	50	50	**3**	2	3
DS4	8	8	**2**	1	2
One tank	8	8	**2**	**2**	2
TE	16	16	**1**	**1**	1
Three tanks	13	13	**2**	**2**	2
DS1+DS4+TE	850	850	–	–	1
DS1+TE	680	680	–	–	1

Number of results for different approaches. Q stands for quantum experiments, and C for experiments on classical computers. Exp. is the expected number of faults

the quantum version of the respective algorithm scales much better than the classical version. For QAOA, the difference between runs with the least number of components and those with the highest number is 1.17 for the quantum case and 14.34 for the classical case. For Grover's algorithm, the difference is even larger: 14.79 for the quantum case and 156.53 for the classical case.

The results on the IBM quantum computer are hard to compare to classical diagnosis algorithms, which are running on local personal computers. This is due to the following reasons: (i) Traditional fault diagnosis algorithms work either directly on the logical knowledge base (Reiter 1987; Feldman et al. 2010) or work through compilation techniques that encode fault diagnosis as a SAT problem (Metodi et al. 2014). However, the encoding required for computation on a quantum computer is different (Diedrich et al. 2016; Perdomo-Ortiz et al. 2019; Ding et al. 2024). (ii) Because of the different encoding, the absolute runtime is not comparable. The computational complexity of the traditional fault diagnosis algorithms is greater than polynomial, depending on the actual diagnosis algorithm (Rodler 2022). However, an intuition of the scaling of traditional fault diagnosis algorithms was shown by Kurtoglu et al. (2009) and Poll et al. (2010) as part of fault diagnosis competitions. Another study of the runtime of diagnosis algorithms was performed by Nica et al. (2013).

5 Discussion

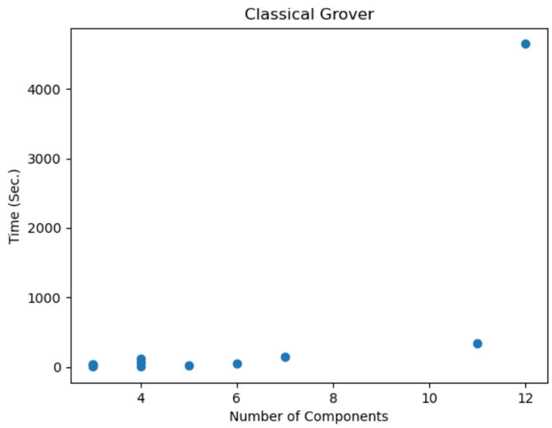
Solving industrial fault diagnosis problems on quantum computers is a promising approach. We have shown how typical examples of cyber-physical systems can be correctly diagnosed using quantum computers (RQ1). We have used the Grover algorithm and a QAOA approach for diagnosis. While the Grover algorithm is faster and can solve larger problems, it is less accurate than the QAOA approach. Conversely, QAOA provides the additional advantages that it is able to

Table 5 Experimental results

System	Grover (Q)	Grover (C)	QAOA (Q)	QAOA (C)
Example 1	1.781 s	0.039 s	3161.565 s	1.007 s
Example 2	1.583 s	0.113 s	2244.280 s	2.098 s
DS1	1.711 s	0.144 s	–	–
DS2	1.517 s	0.013 s	1947.786 s	1.047 s
DS3	1.589 s	0.050 s	2951.424 s	10.810 s
DS4	1.551 s	0.016 s	2993.269 s	11.072 s
One tank	1.529 s	0.037 s	2485.160 s	0.264 s
TE	1.524 s	0.009 s	2254.813 s	3.290 s
Three tanks	1.553 s	0.063 s	2609.910 s	0.731 s
DS1+DS4+TE	23.798 s	4.649 s	–	–
DS1+TE	9.954 s	0.338 s	–	–

Runtime for different approaches. Q stands for quantum experiments, and C for experiments on classical computers

(a) Runtime for running the Grover algorithm on a classical PC



(b) Runtime on a quantum computer using Grover's algorithm

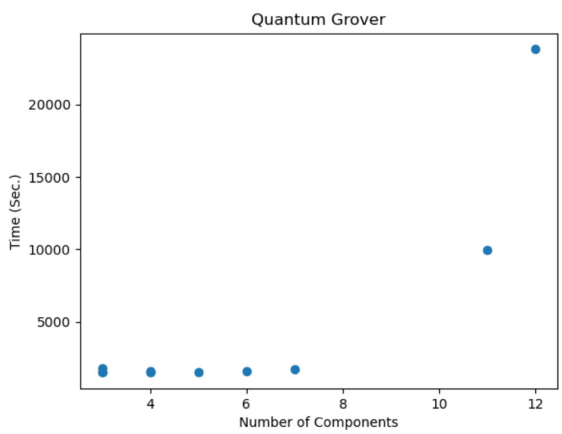
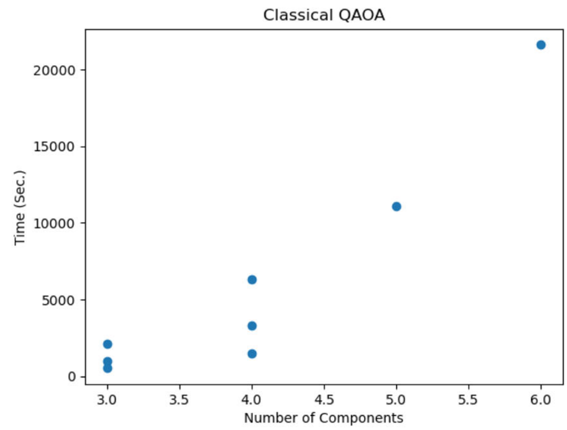


Fig. 2 Experiments using Grover

(a) Runtime for running the QAOA algorithm on a classical PC



(b) Runtime on a quantum computer using QAOA

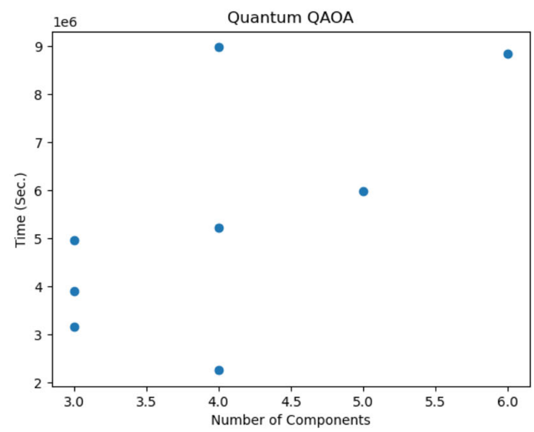


Fig. 3 Experiments using QAOA

find diagnoses with more than one component and that it could be run on quantum annealers as it uses a QUBO formulation. Our method holds points two and three posed by Aaronson and Chen (2016) in that our results are predicted in theory and that the consistency test has passed. As for his point one, given our sampling size, we cannot definitely prove that the quantum/classical gap increases exponentially with instance size.

Apart from the quantum computation, we have also demonstrated how quantum diagnosis is combined with an approach for automated model learning to learn propositional logic system descriptions from process data and test whether a diagnosable system description was learned (RQ2).

There are a couple of limitations to our approach: (i) Using the IBM Falcon processor entails a certain amount of classical computation at each step. This cannot be removed and thus dilutes the empirical results. (ii) The number of qubits is insufficient for truly large experiments. It has been shown, however, that many typical applications can already be solved with the limited number of qubits. (iii) In the implementation, both quantum diagnosis algorithms qDiagCPS_Grover and qDiagCPS_QAOA are neither sound nor complete (Balzereit and Niggemann 2021). Using an oracle and logical constraints, future work will involve making at least one of the algorithms complete. (iv) qDiagCPS_Grover performs far worse than qDiagCPS_QAOA. This is mainly due to the computation of the sorted result list, in which the correct solutions exist, but which are often masked through the existence of a large number of non-optimal solutions. Running the algorithms on larger systems may provide more insight into why the ranking of the solutions is thus far unsatisfactory. (v) Unrelated to the quantum aspects, our approach has

Table 6 Comparison of the average execution time between experiments with minimal number of components and experiments with maximum number of components, as well as the quotient between the maximum and the minimum

	QAOA (Q)	QAOA (C)	Grover (Q)	Grover (C)
Mean min ($\| COMPS \|$)	2531.50 s	0.772 s	1.609 s	0.0297 s
Mean max ($\| COMPS \|$)	2951.424 s	11.072 s	23.798 s	4.649 s
Quotient	1.17	14.34	14.79	156.53

A larger quotient indicates worse scaling performance

two further limitations: The residual generation function $h(\cdot)$ Eq. (2) needs to be determined manually, and the model is not automatically proved to be correct. Both of these limitations will be the subject of future work. Another limitation, especially for the QAOA approach, was system size (number of variables) such that it could be compiled within the 27 qubits that the Falcon processor provides. QAOA is implemented through the use of helper variables. But each variable uses at least one qubit. As such, limitations are evident in the number of helper variables as well as the problem size itself.

Overall, in this article, we have demonstrated the applicability of a practical and scalable approach for diagnosing cyber-physical systems leveraging quantum computers. While quantum computation is still taking longer in absolute numbers compared to local execution on a classical PC, it is expected that this gap will decrease in future quantum architectures.

6 Conclusion

In this article, we have presented a three-pronged solution to perform fault diagnosis for cyber-physical systems in industrial production contexts. We have presented the quantum algorithms qDiagCPS_Grover and qDiagCPS_QAOA, which calculate minimum cardinality diagnoses solving an NP-complete MaxSAT problem. The empirical evaluation has shown that these algorithms scale better on a quantum computer than on a classical PC. However, both algorithms also show some limitations as was presented in Section 5. While the Grover algorithm generates more solutions and needs some post-processing to obtain the minimum cardinality diagnoses, it is much faster in absolute terms than the QAOA approach. We have also presented the algorithm GTSD to determine a system description automatically from process data using a generate-and-test approach. Together with the developed quantum algorithms, GTSD provides a novel practical diagnosis methodology for cyber-physical systems.

Acknowledgements This work was developed within the Fraunhofer Sonderprogramm Quantum Now.

Author Contributions A.D. prepared the figures and performed the experiments. All authors wrote and reviewed the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The data will be made available upon reasonable request. Code for fault diagnosis is available at: https://github.com/DaemonNet

Declarations

Competing Interests The authors declare no competing interests.

References

Aaronson S, Chen L (2016) Complexity-theoretic foundations of quantum supremacy experiments. arXiv:1612.05903

Ajagekar A, You F (2022) New frontiers of quantum computing in chemical engineering. Korean J Chem Eng:1–10

Akaike H (1998) Information theory and an extension of the maximum likelihood principle, 199–213

Balzereit K, Diedrich A, Ginster J, Windmann S, Niggemann O (2021) An ensemble of benchmarks for the evaluation of AI methods for fault handling in CPPS. In: 19th IEEE international conference on industrial informatics

Balzereit K, Niggemann O (2021) Sound and complete reconfiguration for a class of hybrid systems. In: Proceedings of the 32nd international workshop on principle of diagnosis, Hamburg, Germany, pp 13–15

Bian Z, Chudak F, Israel RB, Lackey B, Macready WG, Roy A (2016) Mapping constrained optimization problems to quantum annealing with application to fault diagnosis. Frontiers in ICT, 14

Bisshop J (2019) Optimization modeling

Bittner B, Bozzano M, Cimatti A, Gario M, Tonetta S, Vozarova V (2022) Diagnosability of fair transition systems. Artif Intell 309:103725

Bochman A (2021) A logical theory of causality

Borutzky W (2020) A hybrid bond graph model-based-data driven method for failure prognostic. Procedia Manuf 42:188–196

Brown AL (1974) Qualitative knowledge, causal reasoning, and the localization of failures. MIT

Chao MA, Kulkarni C, Goebel K, Fink O (2022) Fusing physics-based and deep learning models for prognostics. Reliab Eng Syst Safety 217:107961

De Kleer J, Brown JS (1984) A qualitative physics based on confluences. Artif Intell 24(1–3):7–83

De Kleer J, Williams BC (1987) Diagnosing multiple faults. Artif intell 32(1):97–130

Diedrich A, Niggemann O (2022) On residual-based diagnosis of physical systems. Eng Appl Artif Intell 109:104636. https://doi.org/10.1016/j.engappai.2021.104636

Diedrich A, Deutschmann P, Junker, C (2022) Service-navigator - a Bayesian assistance system for diagnosing industrial production systems. In: 2022 5th IEEE International conference on industrial cyber-physical systems (ICPS)[submitted], IEEE

Diedrich A, Feldman A, Perdomo-Ortiz A, Abreu R, Niggemann O, de Kleer J (2016) Applying simulated annealing to problems in model-based diagnosis

Diedrich A, Maier A, Niggemann O (2019) Model-based diagnosis of hybrid systems using satisfiability modulo theory. In: Proceedings of the thirty-third AAAI conference on artificial intelligence (AAAI-19)

Diedrich A, Moddemann L, Niggemann O (2024) Learning system descriptions for cyber-physical systems. In: Proceedings of 12th IFAC symposium on fault detection, supervision and safety for technical processes

Ding J, Spallitta G, Sebastiani R (2024) Effective prime factorization via quantum annealing by modular locally-structured embedding. Sci Rep 14(1):3518

Ehrhardt J, Ramonat M, Heesch R, Balzereit K, Diedrich A, Niggemann O (2022) An AI benchmark for diagnosis. Reconfiguration & planning, IEEE INDIN

Fei X, Zhao H, Zhou X, Zhao J, Shu T, Wen F (2024) Power system fault diagnosis with quantum computing and efficient gate decomposition. Sci Rep 14

Feldman A, Provan G, Van Gemund A (2010) Approximate model-based diagnosis using greedy stochastic search. J Artif Intell Res 38:371–413

Feldman A, de Kleer J, Matei I (2022) A quantum algorithm for computing all diagnoses of a switching circuit. arXiv:2209.05470

Forbus KD (1984) Qualitative process theory. Artificial intelligence 24(1–3):85–168

Gao Z, Cecati C, Ding SX (2015) A survey of fault diagnosis and fault-tolerant techniques—part i: fault diagnosis with model-based and signal-based approaches. IEEE Trans Industr Electron 62(6):3757–3767

Grastien A (2014) Diagnosis of hybrid systems with SMT: opportunities and challenges. In: Proceedings of the twenty-first European conference on artificial intelligence, IOS Press, pp 405–410

Grover LK (1997) Quantum mechanics helps in searching for a needle in a haystack. Phys Rev Lett 79(2):325

Ignatiev A, Morgado A, Weissenbacher G, Marques-Silva J, RAS IS (2019) Model-based diagnosis with multiple observations. In: IJCAI, pp 1108–1115

Jaber A, Zhang J, Bareinboim E (2019) Causal identification under Markov equivalence. In: Twenty-eighth international joint conference on artificial intelligence

Jung D (2022) Automated design of grey-box recurrent neural networks for fault diagnosis using structural models and causal information. In: Learning for dynamics and control conference, PMLR, pp 8–20

Jung D, Sundström C (2017) A combined data-driven and model-based residual selection algorithm for fault detection and isolation. IEEE Trans Control Syst Technol 27(2):616–630

Jung D, Sundström C (2019) A combined data-driven and model-based residual selection algorithm for fault detection and isolation. IEEE Trans Control Syst Technol 27(2):616–630. https://doi.org/10.1109/TCST.2017.2773514

Khan AS, Khan AQ, Iqbal N, Sarwar M, Mahmood A, Shoaib MA (2020) Distributed fault detection and isolation in second order networked systems in a cyber-physical environment. ISA Trans 103:131–142

Khorasgani H, Biswas G, Jung D (2019) Structural methodologies for distributed fault detection and isolation. Appl Sci 9(7):1286

Kolb S, Teso S, Passerini A, De Raedt L (2018) Learning SMT (LRA) constraints using SMT solvers. In: IJCAI international joint conference on artificial intelligence, vol 2018, pp 2333–2340. ijcai.org

Kurtoglu T, Narasimhan S, Poll S, Garcia D, Kuhn L, de Kleer J, van Gemund A, Feldman A (2009) First international diagnosis competition-dxc'09. Proc DX 9:383–396

Leipold H, Spedalieri FM, Rieffel E (2022) Tailored quantum alternating operator ansätzes for circuit fault diagnostics. Algorithms 15(10):356

Leipold H, Spedalieri FM (2022) Quantum annealing with special drivers for circuit fault diagnostics. arXiv:2203.09560

Matei I, Zhenirovskyy M, de Kleer J, Feldman A (2019) A hybrid qualitative and quantitative diagnosis approach. In: Annual conference of the PHM society, vol 11

Merkelbach S, Diedrich A, Niggemann O (2024) Towards the generation of models for fault diagnosis of CPS using VQA models. ML4CPS 2024 – Machine Learning for Cyber Physical Systems Conference

Metodi A, Stern R, Kalech M, Codish M (2014) A novel SAT-based approach to model based diagnosis. J Artificial Intell Res 51:377–411

Mirkarimi P, Callison A, Light L, Chancellor N, Kendon V (2022) Comparing the hardness of max 2-SAT problem instances for quantum and classical algorithms. arXiv:2206.06876

Moddemann L, Steude HS, Diedrich A, Niggemann O (2024) Discret2di - deep learning based discretization for model-based diagnosis. In: Proceedings of 12th IFAC symposium on fault detection, supervision and safety for technical processes

Muškardin E, Pill I, Wotawa F (2020) CatIO-a framework for model-based diagnosis of cyber-physical systems. In: International symposium on methodologies for intelligent systems, Springer, pp 267–276

Nica I, Pill I, Quaritsch T, Wotawa F (2013) The route to success—a performance comparison of diagnosis algorithms. In: Twenty-third international joint conference on artificial intelligence, Citeseer

Nielsen EK, Gofuku A, Zhang X, Ravn O, Lind M (2020) Causality validation of multilevel flow modelling. Comput Chem Eng 140:106944

Niggemann O, Diedrich A, Kühnert C, Pfannstiel E, Schraven J (2021) A generic digital-twin model for artificial intelligence applications. In: 2021 4th IEEE International conference on industrial cyber-physical systems (ICPS), IEEE, pp 55–62

Pearl J (2009) Causality

Perdomo-Ortiz A, Fluegemann J, Narasimhan S, Biswas R, Smelyanskiy VN (2015) A quantum annealing approach for fault detection and diagnosis of graph-based systems. European Phys J Special Topics 224(1):131–148

Perdomo-Ortiz A, Feldman A, Ozaeta A, Isakov SV, Zhu Z, O'Gorman B, Katzgraber HG, Diedrich A, Neven H, de Kleer J et al (2019) Readiness of quantum optimization machines for industrial applications. Phys Rev Appl 12(1):014004

Perdomo-Ortiz A, Feldman A, Ozaeta A, Isakov SV, Zhu Z, O'Gorman B, Katzgraber HG, Diedrich A, Neven H, de Kleer J et al (2017) On the readiness of quantum optimization machines for industrial applications. arXiv:1708.09780

Pill I, Wotawa F (2018) Fault detection and localization using modelica and abductive reasoning. Diagnosability, Secur Safety Hybrid Dyn Cyber-Phys Syst:45–72

Poll S, de Kleer J, Feldman A, Garcia D, Kurtoglu T, Narasimhan S (2010) Second international diagnostics competition–dxc'10. In: Proceedings of the 21st international workshop on principles of diagnosis
Reiter R (1987) A theory of diagnosis from first principles. Artif Intell 32(1):57–95
Rodler P (2022) How should I compute my candidates? A taxonomy and classification of diagnosis computation algorithms. In: 33rd International workshop on principle of diagnosis – DX 2022
Singh NK, Borrok DM (2019) A Granger causality analysis of groundwater patterns over a half-century. Sci Rep 9(1):1–8
Stern R, Kalech M, Elimelech O (2014) Hierarchical diagnosis in strong fault models. In: Twenty fifth international workshop on principles of diagnosis
Sugihara G, May R, Ye H, Hsieh C-H, Deyle E, Fogarty M, Munch S (2012) Detecting causality in complex ecosystems. Science 338(6106):496–500
Wismüller A, Dsouza AM, Vosoughi MA, Abidin A (2021) Large-scale nonlinear granger causality for inferring directed dependence from short multivariate time-series data. Sci Rep 11(1):1–11
Yuan Y, Tang X, Zhou W, Pan W, Li X, Zhang H-T, Ding H, Goncalves J (2019) Data driven discovery of cyber physical systems. Nat Commun 10(1):1–9
Yucesan YA, Dourado A, Viana FA (2021) A survey of modeling for prognosis and health management of industrial equipment. Adv Eng Inform 50:101404