TOWARDS THE GENERATION OF MODELS FOR FAULT DIAGNOSIS OF CPS USING VQA MODELS

Silke Merkelbach[1], **Alexander Diedrich**[2], **Sebastian von Enzberg**[3], **Oliver Niggemann**[2], **and Roman Dumitrescu**[1]
[1]Fraunhofer Institute for Mechatronic Systems Design IEM, Paderborn, Germany
{firstname.lastname}@iem.fraunhofer.de
[2]Helmut-Schmidt-University, Hamburg, Germany
{firstname.lastname}@hsu-hh.de
[3]Hochschule Magdeburg-Stendal, Magdeburg, Germany
sebastian.von.enzberg@h2.de

ABSTRACT

In many use cases cyber-physical systems are employed to produce products of small batch sizes as efficiently as possible. From an engineering standpoint, a major drawback of this flexibility is that the architecture of the cyber-physical system may change multiple times over its lifetime to accommodate new product variants. To keep a cyber-physical system working normally it has become common to employ fault diagnosis algorithms. These algorithms partly rely on physical first-principles models that need to be updated when the architecture of the system changes which usually has to be done manually. In this article we present a practical approach to obtain such a first-principles model through evaluating piping and instrumentation diagrams (P&IDs) with visual questions answering (VQA) models. We demonstrate that it is possible to leverage VQA models to construct physical equations which are a preliminary stage for the creation of models suitable for fault diagnosis. We evaluate our approach on OpenAIs GPT-4 Vision Preview model using a P&ID we created for a benchmark water tank system. Our results show that VQA models can be used to create physical first-principles models.

Keywords Visual Question Answering · Large Language Models · Fault Diagnosis · First-Principles Models · Application of LLMs

1 Introduction

Cyber-physical systems aim to perform some function towards achieving a certain goal. In the area of cyber-physical production systems this goal is often the efficient production of goods at high volume and low costs. Unfortunately, some of the components of a cyber-physical system may break down over time. For example, through wear and tear valves may get stuck, pipes may get leaks, and electronic components may get damaged. In many of these cases production has to be stopped until the defect can be mitigated [11]. Such mitigation is done through the early detection of faults using automated fault diagnosis methods. The goal of the research field of fault diagnosis is to provide algorithms and automated methods to find faults as soon as they occur and optimally to suggest mitigation strategies [3].

The challenge is that usually faults are diagnosed using a model-based approach. But obtaining the correct models that are also adaptive to potential changes to the cyber-physical system over its lifetime is often very expensive. In the past, these models had to be created by experts and needed to be adapted to different operating modes (i.e. ramp-up, production of different products etc.), a multitude of different product variants, and to changes in architecture [25]. Such manual adaptations were usually prohibitively expensive and limited adoption of diagnosis methods. Nowadays, several approaches exist that attempt to learn those models automatically [12, 10, 34, 33, 16] through statistical, control theoretic, and machine learning approaches. But learning models automatically from data always has the drawback that already known facts that can easily be described by experts, may be unaccounted for. In addition, some connections and rare events may not be covered by the learned model if they do not occur in the training data. It is thus highly relevant to develop approaches that take expert knowledge into account in order to automatically create suitable models for diagnosis [35].

Usually, the models needed for diagnosis tasks are referred to as system descriptions (SD). System descriptions can be specified through propositional logic, predicate logic, or even structural equations [13]. They usually express functional dependencies between components, signals and/or variables. In classical consistency-based diagnosis [8] SDs are described using propositional logic. But previous work found that obtaining propositional logic SDs automatically from data has several drawbacks [12, 10]. Others create system descriptions using structural equation models [13, 15], but those require a significant amount of expert knowledge. The challenge we address in this article is to create SDs that capture the physical relationships and dependencies within a cyber-physical system out of piping and instrumentation diagrams (P&IDs). P&IDs are used in the process industry to visualize processes with fluids and gases. We want to use an image with a P&ID as basis to obtain a model that tells us that if a valve is stuck, we will see that the flow through some pipe may not change as expected.

In this article we present a method that uses large language models with vision capabilities, also known as visual question answering (VQA) models, to create physical models from given images containing P&IDs. We propose a two-step solution where we first create a table with connections between the elements in the image and afterwards derive physical equations from it. The resulting model can in turn be used to create analytical redundancy relations (ARRs) or structural models to compare the predicted system behaviour to actual behaviour. Discrepancies between predictions and observations can then be used for diagnosis.

With our presented solution it is possible to automatically create physical first-princples models from P&IDs as basis for system descriptions and, together with actual data, perform diagnosis of cyber-physical production systems from process industry. Overall, our approach shows how to leverage large-language models capable of processing image data to diagnose adaptive and constantly changing system architectures of cyber-physical production systems.

Our contribution is the following: We present a novel approach to generate physical first-principles models of cyber-physical production systems from P&IDs by leveraging a VQA model, namely ChatGPT 4 Vision Preview [32].

We validate our approach in experiments on a P&ID we created for the four-tank system from the benchmark presented by Balzereit et al. [2]. The P&ID uses the symbols of the standard DIN EN ISO 10628-2 [9].

2 Related Work

The idea of identifying causal dependencies for physical modelling was introduced by the works of Forbus [14] and de Kleer et al. [7]. Recently, Nielsen et al. [30] 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. [19] have presented an approach to improve causal reasoning and modelling, and investigated the use of partial ancestral graphs. Chao et al. [6] 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. Cao et al. [5] and also Vokuvić [36] provide a review of causal discovery for industrial processes. They identify that, among several other methods, Granger Causality is a useful method to obtain causal relationships. But these do not correspond to physical models. Frisk et al. [16] have presented work to extract analytical redundancy relations automatically, without using large-language models. But these do not work with image data. An active research field between causality research and fault diagnosis are bondgraph approaches. Gao et al. [17] provide a good overview over bondgraphs and other modelling and model-based diagnosis methods. Recently, Borutzky [4] and Khan et al. [24] have introduced new residual-based approaches for fault diagnosis. In consistency-based diagnosis Matei et al. [28] have published articles about diagnosing physical systems using differential equations. Physical system diagnosis has also recently been addressed by Muškardin et al. [29] and by Yucesan et al. [38]. Kolb et al. [26] have presented a method to learn satifiability modulo theory (SMT) expressions as system models. The closest approach to our method was introduced by Krysander and Nyberg [27] and Gelso et al. [18] who presented the structural analysis of tank systems for fault diagnosis. We conclude that, while many works are available that require first-principles models, few works express the need to learn models automatically. And none do so in the context of fault diagnosis through the use of large-language models.

Large language models have not been broadly adopted by the fault diagnosis community yet. Kang et al. [20] have presented the usage of large language models for software fault localization. The same is true for the work of Wu et al. [37]. Balhorn et al. [1] have used large language models to correct possibly faulty P&ID diagrams, but have not attempted to generate any kinds of models from the diagrams. Ogundare et al. [31] have analysed the resilience of large language models to create system models. But their models are limited to single equations that have no automatic dependencies between each other. Kato et al. [23, 22, 21] extracted equations from a large number of scientific documents and created physical models out of them by judging their equivalence using a pre-trained large language model and defining requirements the physical model has to meet. But they neither use P&IDs nor do they create models for fault diagnosis.

3 Creating Physical First-principles Models using VQA Models

Our method to generate physical models out of P&IDs using VQA models consists of two steps: i) Extracting the structure of the system from the P&ID image data and ii) Creating physical first-principles models. We choose a two-step solution because it proved to be helpful for large language models if the problem is decomposed into smaller pieces [39]. In the first step, we input an image with a P&ID and a prompt including context and task into a VQA model, getting the structure of the system in tabular form as output. In the second step, we input the generated table with the system structure, again the P&ID, and a prompt including assumptions for and context about the resulting model. An overview of our approach in shown in Figure 1.

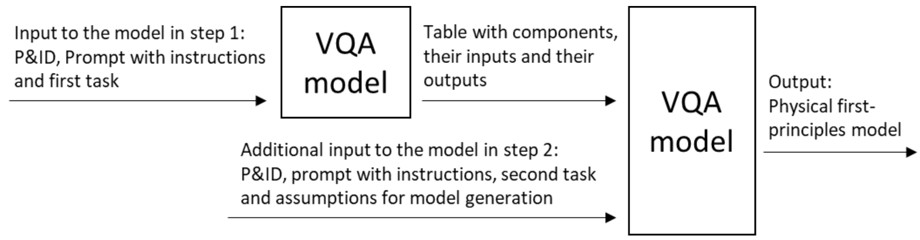

Figure 1: Two-step method to generate physical first-principles models for diagnosis out of P&IDs and additional, textual information

3.1 Extract the structure of the system from the P&ID (step 1)

In this step we generate the structure of the system with its components, their inputs, and their outputs from a P&ID. The P&ID image shows symbols according to DIN EN ISO 10628 [9]. The generated structure is a table that describes the adjacency of the components within the P&ID in electronically usable form. I.e. each row within the table shows a specific component and how it is connected to other components. By creating the table with the components and their connections first, the model has to focus on only one task instead of doing multiple tasks at a time which proved to create better results in the past when using large language models [39]. Our input to the VQA model, besides the P&ID image, are instructions describing the context, and a user message with the concrete task.

The instructions are optional parameters to the GPT-4 model and describe the role the model should take. We specified that the model should take the role of an engineer and included information about the structure of the image and how it should be read (i.e. that it should follow arrows and read from left to right, for example). We also specified the output format in the form of a comma separated values (.csv) file. The user message contains the concrete task that should be fulfilled by the model and is the actual prompt that most people know from using large-language models. It specifies the design of the table, to output the term 'unclear' if components could not be recognized, and to output only the table without any further explanations.

3.2 Create physical first-principles model from the connection table generated in the first step (step 2)

The second step of our method generates the physical model consisting of modelling equations for the identified components out of the following inputs: the connection table which was created in step 1, again the P&ID, the instructions, and the user message with the task and assumptions for model generation. The connection table from step 1 contains the connections between the components and is used to support the model building process. The P&ID is given to the VQA model as reference in case that relevant information is missing in the textual description. The instructions again contain the role the VQA model should take. However, we changed the instructions to include the different input that is used in this step. The user message contains the concrete task that should be fulfilled by the VQA model and specifies the type of model that is needed for the given use case, such as a dynamical model. In addition, the user message includes assumptions that should be made for the creation of the physical first-principles model. The assumptions involve information about the geometry of the system and other information regarding the system, such as insulation or energy losses. They also involve some equations which should be used for modeling. The equations are given to the VQA model to be less dependent on the training data the model has seen so far.

4 Evaluation

To evaluate our approach we did one experiment for the first step and one for the second step. In the following we first describe the design of experiment, then present the results and provide a brief discussion about the limitations of our approach.

4.1 Experiment Design

We evaluate our approach using OpenAI's gpt-4-1106-vision-preview model [32] as VQA model which was accessed via the API interface using Python. As an application example, we took the four-tank-system S3 as presented by Balzereit et al. [2]. To make sure that ChatGPT has not seen the image before, we created a P&ID describing the system according to DIN EN ISO 10628 as shown in Figure 2. The system contains four tanks, seven valves, and sensors measuring flow and level while the tanks and valves have unique names. For simplicity, we ignore influences of temperature and assume it is an ideal system with no energy losses. The system was chosen since the connections are challenging through the separation of fluids after Tank 0, the bypass leading through Valve 3, and the three volume flows which lead to Tank 3.

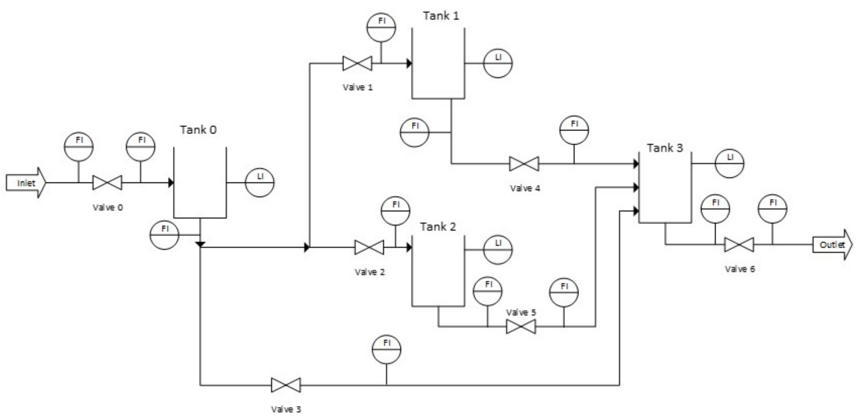

Figure 2: The P&ID of the four-tank system which was used in our experiments

We evaluated both steps of our approach in a separate experiment to better understand where potential problems might occur. This means we took a correct version of the table from step one as input for step 2 instead of using the real output from step 1. To keep the VQA model as deterministic as possible, we used the following parameters provided by OpenAI for prompting via the API interface in Python: seed=42, temperature=0, top_p=0.1, frequency_penalty=0, and presence_penalty=0. We repeated every experiment 100 times to get an idea of possible results and to make sure that valid ones occur more often than wrong ones. To test the reliability of the model, we included the order to write 'unclear' if it is not sure of the result. We used the following prompts in our experiments:

Step 1:
System instructions: "You are an engineer who analyzes piping and instrumentation diagrams. Your task is to identify all elements in the image and describe their input and output connection in tabular form. In the image, there is a piping and instrumentation diagram according to ISO 10628. In the image, there can be symbols for tanks, valves, pumps and other equipment. They are connected by pipes which are shown as lines with arrows indicating the flow direction. Pipes can have edges and do not always lead from left to right but also vertically. Inputs to the system are marked as arrows on the left side of the image and outputs of the system are arrows that lead towards the right side of the image."

User message: "Describe the elements and their connections in tabular form with the following columns: Element, Inputs from, Outputs to. If you are not sure that the answer is correct, write 'unclear'. Give only the table and no further explanations. Provide the output in csv format."

Step 2:
System Instructions: "You are an engineer who has the job to create physical models for process systems. As input, you get a table with elements of a system and which other elements are connected to them via input and output. In addition, you get an image of the system which shows the elements according to DIN EN ISO 10628. The elements are named according to their function. Unique symbols which show if it is an input, an output, or a state variable are usually used in the equations for variables and parameters."

User Message: "Create a dynamical model with physical equations for the volume flow through the valves and the fluid level in the system described in the following table: \"Element,Inputs from,Outputs to\n Valve 0,Inlet,Tank 0\n Tank 0,Valve 0,Valve 1; Valve 2; Valve 3\n Valve 1,Tank 0,Tank 1\n Tank 1,Valve 1,Valve 4\n Valve 4,Tank 1,Tank 3\n Valve 2,Tank 0,Tank 2\n Tank 2,Valve 2,Valve 5\n Valve 5,Tank 2,Tank 3\n Valve 3,Tank 0,Tank 3\n Tank 3,Valve 3; Valve 4; Valve 5,Valve 6\n Valve 6,Tank 3,Outlet\".\n\nUse the following Assumptions:\n\"The fluid in the system is water\nThe fluid is incompressible\nThere are no energy losses\nThe process is adiabatic\nThe tanks are open\nThe valves are at the bottom level of the tank they get feed from\nThe inlet of the tanks is at the top at the same height as the valve the tank gets its input from\nThe inlet to the system comes from an infinite water reservoir with 1m height above the first valve\nAll tanks have the same diameter at all heights\nPipes have the same cross-sectional area everywhere\nthe cross-sectional area of the outlet of tank 0 is three times larger than the cross-sectional area of the pipes\nThe equation for the fluid levels in the tanks i is \frac{dH_{T_i}}{dt} = \frac{1}{A_{T_i}}(\sum\nolimits Q_in-\sum\nolimits Q_{out})\nThe equation for the flow through the valves j is \dot{Q}_{V_j} = C_{V_j} \cdot a_{V_j} \cdot \sqrt{2gH_{T_k}} with T_k being the tank before the valve\"\n\nIf you are not sure that the answer is correct, write 'unclear'. Create the output in Latex.

Give only the equations in executable latex code and no further explanations. Add a brief description of the used symbols in latex."

Element	Inputs from	Outputs to
Valve 0	Inlet	Tank 0
Tank 0	Valve 0	Valve 1, Valve 2, Valve 3
Valve 1	Tank 0	Tank 1
Tank 1	Valve 1	Valve 4
Valve 4	Tank 1	Tank 3
Valve 2	Tank 0	Tank 2
Tank 2	Valve 2	Valve 5
Valve 5	Tank 2	Tank 3
Valve 3	Tank 0	Tank 3
Tank 3	Valve 3, Valve 4, Valve 5	Valve 6
Valve 6	Tank 3	Outlet

Table 1: Sample solution of step 1 and input to step 2.

4.2 Results of the Experiments

The goal for the large-language model in step 1 was to create a table with eleven rows which equals to the number of components in the system. Table 1 shows the sample solution. In all cases, the format was generated correctly. In two repetitions, the inlet and the outlet were recognized as components. Since they were correctly inserted to the table, we did not consider them in the evaluation. The result was considered correct if the complete content of the cell was correct. If something was missing or if a wrong term was included, we counted it as wrong. In total, there were 16 unique results over all 100 runs. The unique results occurred with the following frequencies: 35, 19, 15, 10, 5, 3, 3x2, and 7x1. The results of the experiment are shown in Table 2. We calculated mean, minimum, maximum, median, and the occurrence of 'unclear' in the wrong predictions over all runs. All of the tanks and valves in the image were recognised correctly in each run. The sensors did not occur in any of the results which means the VQA model was able to extract the important parts of the image. More inputs than outputs were identified correctly but the model added 'unclear' more often for the outputs. This means that the trustworthiness of the model to generate outputs is higher than for inputs since the model told us there was an error instead of predicting a wrong value. Unfortunately, the model was never able to recognise the connection between valve 3 and tank 3 correctly, but at least marked the output of valve 3 as 'unclear' in all 100 runs.

The goal of step 2 was to create an equation for every component in the system out of the eleven components in total. The output of the VQA model consisted of four parts in the majority of the runs, namely equations for the valves, equations for the tanks,

	Mean	Min	Max	Median	'unclear' of wrong prediction
Components	1.00	1.00	1.00	1.00	n.a.
Inputs	0.87	0.64	0.91	0.82	0.08
Outputs	0.76	0.36	0.91	0.68	0.83

Table 2: Results of the experiments for step 1. Mean: part of correctly identified components over all runs; Min: Minimum over all runs; Max: Maximum over all runs; Median: Median over all runs; "'unclear' of wrong prediction": Part of the wrong predictions that were marked as unclear (over all runs).

description of the parameters, and, in some cases, additional hints to take into account when using the model. We focused on the validation of the equations, since they would be used further to create models suitable for diagnosis. An equation was considered correct, if all the variables and parameters were included in the correct way. The results showed minor deviations in notation, such as writing $\dot{Q}$ or Q, different words for the reservoir at the inlet of the system (T_{inlet}, T_{res} or $T_{reservoir}$), and using Q_{in} instead of Q_{V_0}. Since these deviations do not affect the performance of the physical model as long as they are consistent over the complete model, they were accepted as correct versions of the model. There were 19 unique results within the 100 runs which occurred at the following frequencies: 52, 13, 8, 2x4, 3, 3x2, and 10x1. The mentioned minor deviations were counted as different unique results. In this step, the word 'unclear' did not occur at all. In total, 99% of the equations were correct while 91 runs generated the physical model completely correct.

4.3 Discussion of the Results

The experiments show the potential of VQA models for the creation of physical first-principles models out of P&IDs. While the second step indicates good performance, the first step holds some potential for improvement. In general, the result is strongly influenced by the data the VQA model has seen during training which makes the results less predictable when applying the proposed method to other systems. Especially since the training data for GPT-4 is not published. The fact that the VQA model was not able to recognize the connection between valve 3 and tank 3 shows that the VQA model might have seen a similar P&ID before, even though we created it from scratch. We did not validate the whole chain by using the output from step 1 as input to step 2. Instead, we used a known, correct version, such that we could look at the behaviour of the two steps on their own and identify the weaknesses of the approach. Thus, the output of step 2 is independent of the actual output of step 1. So far, we cannot judge to which system size our method scales. We expect that it might become necessary to split larger systems into smaller parts and develop strategies to combine the results. In addition, we only tested the approach with ChatGPT so far. Other systems and VQA models should be tested to achieve a more reliable assessment of the applicability of our method. Another point that might reduce the applicability is the fact, that suitable equations need to be given to the model in advance. An automated method to choose and input the equations is desirable. Before our method can be applied in practice, further investigations for the behaviour with other systems and VQA models need to be made.

5 Conclusion

We presented a method to generate physical first-principles models as basis for models that might be applied in fault diagnosis of cyber-physical systems. Our method consists of two steps and relies on VQA models. The first step is to identify components with their inputs and outputs on a given P&ID and to generate the result in tabular form. The second step is to use this table together with further information about the desired model and to generate the physical first-principles model out of it. The experiments, which were done with a P&ID we created for a benchmark system, showed that our method is able to create a model with some limitations. In the first step, 87% of the components and connections were recognized correctly and in the second step, 99% of the equations were created correctly. Future research should focus on improving the detection of components and their connections in P&IDs, testing the approach with more systems and VQA models, and adding steps to generate models that can directly be used for fault diagnosis.

References

[1] L. S. Balhorn, M. Caballero, and A. M. Schweidtmann. Toward autocorrection of chemical process flowsheets using large language models. *arXiv preprint arXiv:2312.02873*, 2023.

[2] K. Balzereit, A. Diedrich, J. Ginster, S. Windmann, and O. Niggemann. An ensemble of benchmarks for the evaluation of ai methods for fault handling in cpps. In *19th IEEE International Conference on Industrial Informatics*, 11 2021.

[3] K. Balzereit and O. Niggemann. Autoconf: A new algorithm for reconfiguration of cyber-physical production systems. *IEEE Transactions on Industrial Informatics*, 2022.

[4] W. Borutzky. A hybrid bond graph model-based-data driven method for failure prognostic. *Procedia Manufacturing*, 42:188–196, 2020.

[5] L. Cao, J. Su, Y. Wang, Y. Cao, L. C. Siang, J. Li, J. N. Saddler, and B. Gopaluni. Causal discovery based on observational data and process knowledge in industrial processes. *Industrial & Engineering Chemistry Research*, 61(38):14272–14283, 2022.

[6] M. A. Chao, C. Kulkarni, K. Goebel, and O. Fink. Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, 217:107961, 2022.

[7] J. De Kleer and J. S. Brown. A qualitative physics based on confluences. *Artificial intelligence*, 24(1-3):7–83, 1984.

[8] J. De Kleer and J. Kurien. Fundamentals of model-based diagnosis. *IFAC Proceedings Volumes*, 36(5):25–36, 2003.

[9] Deutsches Institut für Normung e.V. (DIN). Din en iso 10628-2: Flow diagrams for process plants - part 2: Graphical symbols. Din standard, 2012.

[10] A. Diedrich, F. Buchholz, and O. Niggemann. Learning a causal system description for diagnosing physical systems. In *Proceedings of the 33rd International Workshop on Principles of Diagnosis, Toulouse, France.*, 2022.

[11] A. Diedrich, P. Deutschmann, and C. Junker. Servicenavigator - a bayesian assistance system for diagnosing industrial production systems. In *2022 5th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)[submitted]*. IEEE, 2022.

[12] A. Diedrich and O. Niggemann. Diagnosing systems through approximated information [submitted]. *13th Annual Conference of the Prognostics and Health Management Society*, 2021.

[13] T. Escobet, A. Bregon, B. Pulido, and V. Puig. *Fault Diagnosis of Dynamic Systems*. Springer, 2019.

[14] K. D. Forbus. Qualitative process theory. *Artificial intelligence*, 24(1-3):85–168, 1984.

[15] E. Frisk and M. Krysander. Residual selection for consistency based diagnosis using machine learning models. *IFAC-PapersOnLine*, 51(24):139–146, 2018.

[16] E. Frisk, M. Krysander, and D. Jung. A toolbox for analysis and design of model based diagnosis systems for large scale models. *IFAC-PapersOnLine*, 50(1):3287–3293, 2017.

[17] 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*, 62(6):3757–3767, 2015.

[18] E. R. Gelso, S. M. Castillo, and J. Armengol. An algorithm based on structural analysis for model-based fault diagnosis. In *CCIA*, pages 138–147, 2008.

[19] A. Jaber, J. Zhang, and E. Bareinboim. Causal identification under markov equivalence. In *Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[20] S. Kang, G. An, and S. Yoo. A preliminary evaluation of llm-based fault localization. *arXiv preprint arXiv:2308.05487*, 2023.

[21] S. Kato, K. Kanegami, and M. Kano. Processbert: A pre-trained language model for judging equivalence of variable definitions in process models. *IFAC-PapersOnLine*, 55(7):957–962, 2022.

[22] S. Kato and M. Kano. Efficient physical model building algorithm using equations extracted from documents. In *Computer Aided Chemical Engineering*, volume 52, pages 151–156. Elsevier, 2023.

[23] S. Kato, C. Zhang, and M. Kano. Simple algorithm for judging equivalence of differential-algebraic equation systems. *Scientific reports*, 13(1):11534, 2023.

[24] A. S. Khan, A. Q. Khan, N. Iqbal, M. Sarwar, A. Mahmood, and M. A. Shoaib. Distributed fault detection and isolation in second order networked systems in a cyber–physical environment. *ISA transactions*, 103:131–142, 2020.

[25] H. Khorasgani and G. Biswas. Structural fault detection and isolation in hybrid systems. *IEEE Transactions on Automation Science and Engineering*, 2017.

[26] S. Kolb, S. Teso, A. Passerini, and L. De Raedt. Learning smt (lra) constraints using smt solvers. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2018, pages 2333–2340. ijcai. org, 2018.

[27] M. Krysander and M. Nyberg. Fault diagnosis utilizing structural analysis. *CCSSE, Norrköping, Sweden*, 2002.

[28] I. Matei, M. Zhenirovskyy, J. de Kleer, and A. Feldman. A hybrid qualitative and quantitative diagnosis approach. In *Annual Conference of the PHM Society*, volume 11, 2019.

[29] E. Muškardin, I. Pill, and F. Wotawa. Catio-a framework for model-based diagnosis of cyber-physical systems. In *International Symposium on Methodologies for Intelligent Systems*, pages 267–276. Springer, 2020.

[30] E. K. Nielsen, A. Gofuku, X. Zhang, O. Ravn, and M. Lind. Causality validation of multilevel flow modelling. *Computers & Chemical Engineering*, 140:106944, 2020.

[31] O. Ogundare, G. Q. Araya, I. Akrotirianakis, and A. Shukla. Resiliency analysis of llm generated models for industrial automation. *arXiv preprint arXiv:2308.12129*, 2023.

[32] OpenAI. Chatgpt 4 vision preview. version gpt-4-1106-vision-preview. url: https://platform.openai.com/docs/api-reference, 2023.

[33] J. Runge, S. Bathiany, E. Bollt, G. Camps-Valls, D. Coumou, E. Deyle, C. Glymour, M. Kretschmer, M. D. Mahecha, J. Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):2553, 2019.

[34] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science advances*, 5(11):eaau4996, 2019.

[35] L. Von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021.

[36] M. Vuković and S. Thalmann. Causal discovery in manufacturing: A structured literature review. *Journal of Manufacturing and Materials Processing*, 6(1):10, 2022.

[37] Y. Wu, Z. Li, J. M. Zhang, M. Papadakis, M. Harman, and Y. Liu. Large language models in fault localisation. *arXiv preprint arXiv:2308.15276*, 2023.

[38] Y. A. Yucesan, A. Dourado, and F. A. Viana. A survey of modeling for prognosis and health management of industrial equipment. *Advanced Engineering Informatics*, 50:101404, 2021.

[39] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.