Influence Estimation In Multi-Step Process Chains Using Quantum Bayesian Networks

Maximilian Selch[1] Daniel Müssig[2] Albrecht Hänel[3] Jörg Lässig[4] Steffen Ihlenfeldt[5]

Abstract: Digital representatives of physical assets and process steps play a decisive role in analysing properties and evaluating the quality of the process. So-called digital twins acquire all relevant planning and process data, which provide the basis, for example, to investigate path accuracies in manufacturing. Each single process step aims to perform an ideal machining after the specification of a target geometry. However, the practical implementation of a step usually shows deviations from the targeted shape. The machine-learning based method of probabilistic Bayesian networks enables the quality estimation of the holistic process chain as well as improvements by targeted considerations of single steps and influence factors. However, the handling of large-scale Bayesian networks requires a high computational effort, whereas the processing with quantum algorithms holds potential improvements in storage and performance. Based on the issue of path accuracy, this paper considers the modelling and influence estimation for a milling operation including experiments on superconducting quantum hardware.

Keywords: manufacturing; path accuracy; digital twin; quantum circuit; quantum algorithm; Bayesian networks

1 Introduction

In recent years, the design if quantum software as well as the construction of real quantum hardware have experienced remarkable progress, which leads to an increasing awareness in a broad range of fields of application to investigate potential operational areas of quantum technologies. The ambitious roadmaps for the upcoming years of major players in quantum computing like IBM [GFW21] and IonQ [Ch20] underline the current dynamics in the enhancements of quantum technologies.

One of these scopes of application is manufacturing, for which quantum computers may help to improve simulations to design and test products, to analyse and work with new materials

[1] Fraunhofer Institute for Machine Tools and Forming Technology IWU, Department Digital Production Twin, Nöthnitzer Straße 44, 01187 Dresden, Germany maximilian.selch@iwu.fraunhofer.de
[2] Fraunhofer IOSB, branch Advanced System Technologies IOSB-AST, Cognitive Energy Systems, Wilhelmsplatz 11, 02826 Görlitz, Germany daniel.muessig@iosb-ast.fraunhofer.de
[3] Fraunhofer Institute for Machine Tools and Forming Technology IWU, Department Digital Production Twin, Nöthnitzer Straße 44, 01187 Dresden, Germany albrecht.haenel@iwu.fraunhofer.de
[4] Fraunhofer IOSB, branch Advanced System Technologies IOSB-AST, Cognitive Energy Systems, Wilhelmsplatz 11, 02826 Görlitz, Germany joerg.laessig@iosb-ast.fraunhofer.de
[5] Fraunhofer Institute for Machine Tools and Forming Technology IWU, Department Digital Production Twin, Nöthnitzer Straße 44, 01187 Dresden, Germany steffen.ihlenfeldt@iwu.fraunhofer.de

doi:10.18420/inf2022_99

as well as to proceed risk and quality modeling in supply and process chains. One issue of the latter topic is to analyse the accuracy of machining processes, to identify dependency relations between the individual characteristics and to search for the decisive influencing factors, which case deviations of the work piece from a preset target geometry. These considerations enable an estimation of the holistic quality of the holistic process chain as well as improvements by considering single steps in a targeted manner. Probabilistic Bayesian networks have proved as an appropriate methodology to reproduce such characteristics of a system. However, with growing network size and complexity, the time required for computations of the system increases exponentially. Especially for handling large-scale networks, quantum algorithms offer a way of a compact representation and potential computational advantages.

After providing an overview of some approaches for quality modeling and influence detection in manufacturing as well as related work in the context of Quantum Bayesian networks, we briefly illustrate the methodology using conventional bit-based computer architecture and how to transfer it to a quantum algorithm. Thereby, we supplement the modeling of Quantum Bayesian networks by an estimator for strengths of influences with the help of SWAP-tests. Afterwards, the presented methods are implemented on superconducting quantum computers based on an exemplary machining application of a milling operation. We evaluate this scenario for a small and larger scale setup and finally discuss on the results obtained from current quantum hardware.

2 Related work

Various statistical and ML-based approaches have been proposed to proceed influence evaluation, deviation analysis and quality improvement of machining processes. For these purposes, methods like regression and classification models [Bu21], fault tree analysis [YBA18] as well as design of experiment techniques [DDT20] are mainly used. These approaches often require some advanced adjustments of the underlying physical process (e.g. empirical formulas for regression). In contrast, Bayesian networks allow to make such statements without previously describing the physics in-depth.

The construction and calculations of Bayesian networks as quantum computing algorithms bases on a representation technique of conditional probabilities which is referred to as qsample encoding [SP18]. The setup of quantum circuits and complexity analysis of qsample encoding for Bayesian networks are described in [ORR13, YC14].

3 Modeling and influence estimation using Bayesian networks

Bayesian networks are directed acyclic graphs, in which nodes model certain influence factors as random variables. If nodes are dependent on each other, they are connected

by an edge, whose direction characterises the dependency relation (pointing from the influencing factor to the dependent variable). If a node is not dependent on any other factor, the corresponding variable holds a probability distribution indicating the likelihood of different status the variable can accept. For this purpose, both discrete as well as continuous variables can be modeled. The dependency relations between the nodes are represented by conditional probabilities. In case of discrete variables these dependencies are stored in conditional probability tables (CPT). When using continuous values, the nodes are usually assumed as normal random variables. Dependencies are modelled by Gaussian distributions, which contain parent-related parameters [Jo01].

The considerations in this article will focus on Bayesian networks with discrete variables for two reasons. Firstly, the influence analysis investigates the impacts of process factors on the basis of pre-defined status, which requires a discretisation of continuous values at some point of the modelling procedure. Secondly, discrete Bayesian networks are particularly suitable for the transmission to quantum algorithmic. In the target application of influence estimations in multi-step machining processes, Figure 1 represents a network example with different part programs of a process chain. In this case, each of the variables is assumed to be binary with probability of $\mathcal{P}_+$ to hold a predefined tolerance criterion for the manufacturing quality resp. a probability of $\mathcal{P}_-$ to not fulfil the criterion. Exemplary, a deviation from a target position with respect to a certain axis can be considered with a threshold value respresenting the quality criterion.

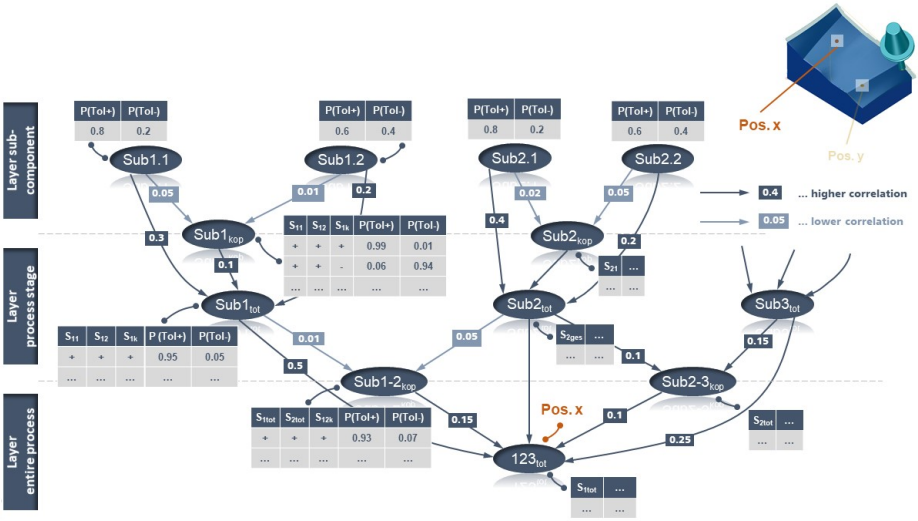

Fig. 1: Example of a Bayesian Network with binary nodes

The structure of a Bayesian network can be obtained by expert knowledge or learned using training data [Jo01]. A Bayesian network with nodes $\mathcal{X} = \{X_1, \ldots, X_m\}$ reflects a unique

joint probability distribution $\mathcal{P}(\mathcal{X})$ which is – in the case of using discrete nodes – given by the product of all conditional probability tables

$$\mathcal{P}(\mathcal{X}) = \prod_{i=1}^{m} \mathcal{P}(X_i | \operatorname{Pa}(X_i)) . \tag{1}$$

In equation (1), $\operatorname{Pa}(X_i)$ describes the set of parent nodes of a variable X_i, i.e. the set of variables on which X_i is dependent on. The single conditional probabilities between dependent nodes are represented as the parameters of the Bayesian network. A parameter

$$\theta_{ijk} = \mathcal{P}(X_i = k | \operatorname{Pa}(X_i) = j) \qquad 1 \leq k \leq s_i, 1 \leq j \leq p_i$$

reflects the probability of a child node X_i to assume if the parent nodes show the status j with s_i depicting the number of different states of X_i and p_i specifying the number of status combinations of $\operatorname{Pa}(X_i)$.

Using classical computing, the parameters of a Bayesian network are commonly learned using the expectation maximization algorithm [PW09]. Given a dataset with $n \in \mathbb{N}$ samples $\mathcal{D} = \{D_1, \ldots, D_n\}$, the goal of the parameter learning is to find the set of parameters $\theta = \{\theta_{ijk}\}$, such that a sample taken from the network matches the data $\mathcal{D}$ best. The Bayesian network can be used to estimate the strength of influences in its dependency relations by analysing the changes in the probability distribution $\mathcal{P}(X_i)$ of a child node X_i when setting evidence to their parent nodes status. The strength of influence of a parent node X_{par} on a child X_{ch} can be described as

$$\mathcal{I}_{X_{\text{par}}}(X_{\text{ch}}) = \frac{1}{\sqrt{2}} \sum_{j=1}^{s_{\text{par}}} \mathcal{P}(X_{\text{par}} = j) \sqrt{\sum_{l=1}^{s_{\text{ch}}} \left(\left[\mathcal{P}\left(X_{\text{ch}} | X_{\text{par}} = j\right) \right]_l - \left[\mathcal{P}(X_{\text{ch}}) \right]_l \right)^2} \tag{2}$$

where s_{par} is defined as the number of different status of X_{par} and s_{ch} as the number of status of X_{ch}. Equation (2) shows the weighted sums of Eucleadian distances of $\mathcal{P}(X_{\text{ch}})$ and $\mathcal{P}\left(X_{\text{ch}} | X_{\text{par}}\right)$ of the different status configurations of the parent node X_{par} – normalised to $\mathcal{I}_{X_{\text{par}}}(X_{\text{ch}}) \in [0, 1]$.

4 Processing Bayesian networks with quantum computing

The calculation of the strength of influence values requires probabilistic inference, i.e. the calculation of posterior distribution of variables when evidence is given by varying the states of parent nodes. This demands a high computational effort in Bayesian networks including a large number of nodes. In the light of the above, this paper focusses on the transfer of this set-up to quantum computing. In a Quantum Bayesian network, each nodes is represented by one or multiple qubits, at which a modelling of s different status of a node requires $\lceil \log_2(s) \rceil$ qubits. Using the technique of qsample encoding, the amplitude vector of a s-qubit quantum state can be used to represent a classical discrete probability distribution. For a probability

distribution of a node X with s different status $\mathcal{P}(X) = p_1, \ldots, p_s,\ (s = 2^n, n \in \mathbb{N})$, measuring the quantum state

$$|\psi\rangle = \sum_{l=1}^{\log_2(s)} \sqrt{p_l}\,|l\rangle \tag{3}$$

in the computational basis is equivalent to sample from $\mathcal{P}(X)$ [SP18]. The encoding of probability distributions for $s = 2$ is done via rotational gates

$$R_Y(\Theta) = \begin{pmatrix} \cos\left(\frac{\Theta}{2}\right) & -\sin\left(\frac{\Theta}{2}\right) \\ \sin\left(\frac{\Theta}{2}\right) & \cos\left(\frac{\Theta}{2}\right) \end{pmatrix}$$

with $\Theta = 2\arcsin(\sqrt{p_2}) =: P_\Theta(p_2)$ to describe $\mathcal{P}(X) = p_1, p_2$ by the probabilities of measuring the state $|0\rangle$ or $|1\rangle$, respectively, of the quantum state $R_Y(\Theta)\,|0\rangle$. Furthermore, to model distributions for $s > 2$, a combination of rotational and controlled rotational is needed to interconnect the required number of qubits to a quantum state representing the node's probability distribution. For example, Figure 2 depicts the circuit to encode $s = 3$ discrete status using the angles

$$\Theta_1 = P_\Theta\left(P\left(|10\rangle\right)\right),\ \Theta_2 = 0, \Theta_3 = P_\Theta\left(\frac{P\left(|01\rangle\right)}{P\left(|00\rangle\right) + P\left(|01\rangle\right)}\right).$$

Thereby, $P\left(|00\rangle\right)$ matches p_1 in $\mathcal{P}(X) = p_1, p_2, p_3$ and $P\left(|01\rangle\right)$ represents p_2 as well as $P\left(|10\rangle\right)$ describes p_3, respectively.

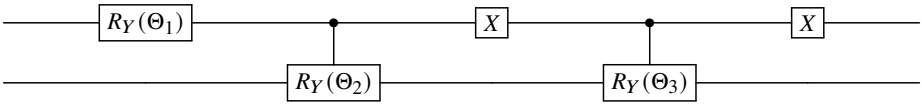

Fig. 2: Quantum circuit to encode a discrete probability distribution with three status

The conditional probability tables of the dependent nodes of the Bayesian network are assembled by a sequence of controlled rotational gates. The controls are the qubits representing the parent nodes and the target is on the child node. Detailed examples of how to proceed these dependency relations in Quantum Bayesian networks are described in [Bo21].

Thus, in comparison with the classical counterpart, Quantum Bayesian networks offer a bunch of advantages in context of memory complexity, itemised:

- compact representation of the network,
- the network's parameters are directly encoded in the qubits representing the nodes,
- logarithmic scaling compared to classical implementation.

Having discussed the build-up of Quantum Bayesian network, a measure for quantify strength of influences analog to the classical way (2) is required. For this purpose, the SWAP-test procedure is suited. The SWAP-test states the certainty of two quantum states to be different from each other. This statement can be transferred to a measure describing the distance between both quantum states. Thereby, a distance of zero corresponds to a probability of 1 of both quantum states to be equal. This probability is given by measurements of ancillary quantum state of the SWAP-test. The other way around, the probability of both quantum states to be different is obtained measuring the ancillary

$$P(\text{anc} = |1\rangle) = \frac{1}{2} - \frac{1}{2} |\langle\psi|\phi\rangle|^2 \tag{4}$$

[Sa20]. Classically, the Euclidean distance between two vectors $\vec{x}$ and $\vec{y}$ representing quantum states calculates as

$$\text{dist}(\vec{x}, \vec{y}) = |\vec{x}, \vec{y}| = \sqrt{(\vec{x} - \vec{y})^2} = \sqrt{|\vec{x}|^2 + |\vec{y}|^2 - 2\langle\vec{x}, \vec{y}\rangle} = \sqrt{2 - 2\langle\vec{x}, \vec{y}\rangle} \tag{5}$$

since vectors representing quantum states are normalised vectors. Comparing both measures, the Euclidean distance (5) correlates positively with the probability (4) of both quantum states to differ from each other. Therefore, to estimate strength of influences in Quantum Bayesian networks, SWAP-tests are proceeded to determine the difference between a specific node with and without giving evidence on parent nodes.

5 Transfer of Bayesian networks to production engineering

In this section, the concept of Quantum Bayesian network is applied to a 3-axis milling operation as a data-driven approach of analysing influence relations between the process factors. The example provides the manufacturing of an aerospace component. The milling operation consists of four processing steps: a face milling, a drilling, a contour milling and end milling operation. The process is visualised in Figure 3 which also shows the assembled number of data records for each single step.

For the setup of Bayesian networks, the following process variables are used

- the axis-specific jerks j_x, j_y, j_z,
- the axial and total position deviations $\Delta_x, \Delta_y, \Delta_z, \Delta_{\text{tot}}$,
- the axis-specific drive currents I_x, I_y, I_z,
- the spindle load L_S,
- the cutting forces according to the Kienzle cutting force model [KV57].

Figure 4 depicts the structure of the corresponding Bayesian network which indicates the dependency relations between the process variables. In [Se21], the results of the strength

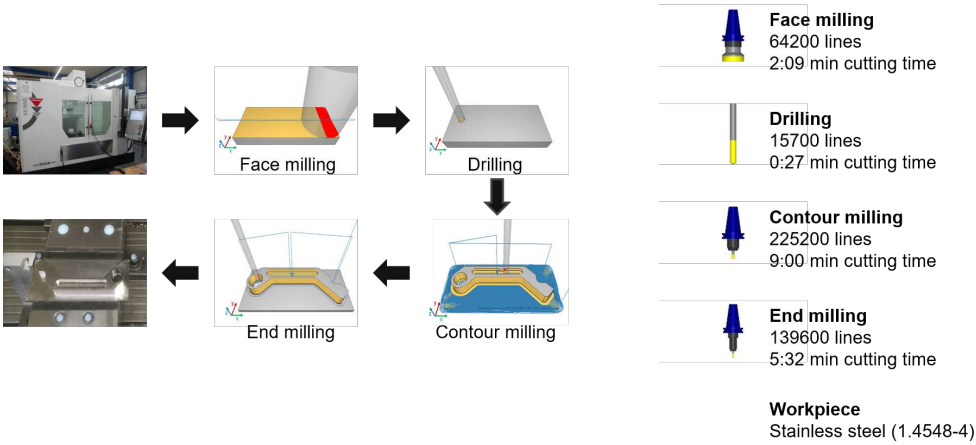

Fig. 3: Manufacturing steps of an aerospace component

of influence estimation via classical Bayesian networks is described. For the evaluation of the quantum computing approach, experiments on the IBM quantum systems are executed using quantum simulators as well as real superconducting quantum hardware devices.

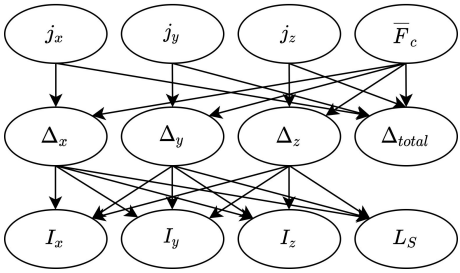

Fig. 4: Structure of the Bayesian network representing the milling application

Quantum Bayesian networks using $s = 2$ and $s = 3$ discrete status for each node were built up. In the $s = 3$ case, only a sub-network of the structure in Figure 4 was implemented in the quantum circuit to be executable at least on IBM's qasm simulator. Figure 5 provides an overview of the calculations for the strength of influence scores to compare the influences of single axial deviations on the spindle load based on (4).

Exemplary, this sub-network is evaluated for the contour milling part which provides the widest variety of machining operations compared to the other sub-processes. In table 1, the strength of influence scores obtained by classical and Quantum Bayesian networks are contrasted. Here, the individual scores can only be compared inside the respective approach, since the positive correlation between the classical and the quantum distance measure does not imply a specific scaling between both scores. Larger values indicate stronger influences

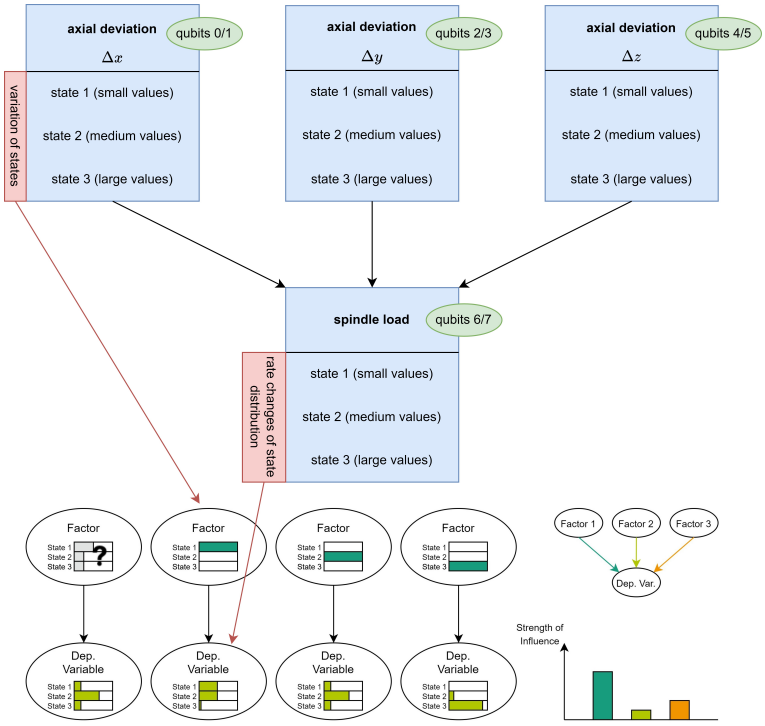

Fig. 5: Setup of the experiments on Quantum Bayesian networks for 3-status nodes

of the respective causal process variables. In both approaches, the influence of the x- and y-axis are dominating the workload. This is in line with the expectations, since these axes correspond to the feed direction of the milling operations.

Approach	Δ_x	Δ_y	Δ_z
Classical Bayesian networks	0.245	0.200	0.161
Quantum Bayesian networks	0.070	0.071	0.059

Tab. 1: Influence scores for process step of contour milling using classical and quantum networks

The execution of the circuits to build up Quantum Bayesian networks and calculating the strengths of influences on today's real quantum computers faces the following device- and circuit-specific problems:

- Real quantum, superconducting devices do not offer an all-to-all connectivity between the physical qubits. To match the circuit with the coupling map of qubits, a transpiling process is required, which increases the depth of the circuit.
- A reduced pool of available quantum gates on the device leads to decompositions of operators, further increasing the circuit complexity.

- The feasibility and accuracy of the circuit depends on the fidelity of the quantum operations (errors of quantum operations).
- The complexity of generated quantum circuits (for example of 3-state-nodes: circuit depth of 1174, width of 40 on IBM qasm_simulator).

6 Using Grover Search

Grover Search was initially framed as an algorithm for searching in unstructured databases [Gr96]. However, at its core, it is a special case of an amplitude amplification algorithm, which can be used in Bayesian networks for interference. The original Grover Search starts in a uniform superposition and searches for one bit-string. In the following years, the algorithm was generalized to find more than one bit-string and to start in any given distribution. Gilliam et al. have shown an extension to the algorithm, where we don't even need to know the searched bit-string. Their algorithm needs to know only if it is the smallest or highest value [GWG20].
For the interference step of the Quantum Bayesian Network (QBN) we start with the probability distribution given by the QBN. Afterward, we apply several Grover-Iterations. A Grover-Iteration consists of an oracle and a diffusion operator. The oracle is a unitary diagonal matrix, which multiplies all bit-strings, which we are looking for, with a -1 and everything else with 1. For example, if we are looking for all bit-strings, where the second qubit is 0 in a 3-qubit system, we need to multiply the following bit-strings by -1: `'000'`, `'001'`, `'100'`, `'101'`.
The second step in a Grover iteration is the diffusion operator, which resembles a reflection of all the amplitudes at their mean value. The operator consists of three steps. The first one is the inverse of creating our initial probability distribution, namely the inverse of our QBN. The third step is to execute the QBN in its natural form. And in between, we need to multiply all amplitudes by -1 except the $|0\rangle_n$ state. The complete diffusion operator is illustrated in Fig. 6. After we have all of our tools together we can execute our interference

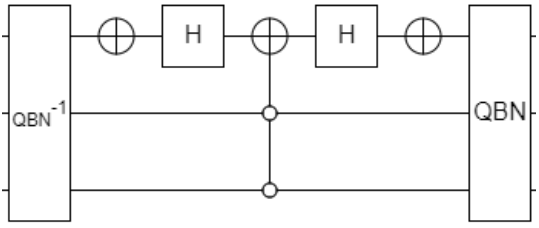

Fig. 6: Diffusor-Operator for a smaller example.

circuit in a hybrid manner. Meaning, that we start with a certain amount of iterations and increase them as we go. We stop, when all of our undesired states are in the region of 0% probability, as Fig. 7 shows. When we compare this approach to the inteference step using a SWAP-Test in Sec. 4 we have a trade off. The implementation with Grover has a high depth,

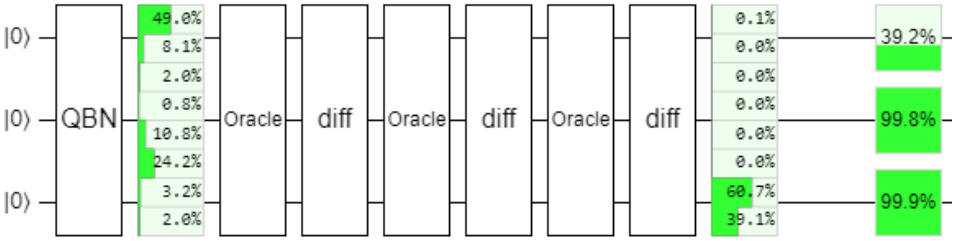

Fig. 7: Grover Serach example for a smaller QBN. The figure shows first the Quantum Bayesian Network as an gate. Afterwards the tool shows the probabilities of all possible states from $|000\rangle$ to $|111\rangle$. Then we apply in this example three Grover iterations, each consisting of the oracle and the diffusion operator. After the iterations we see the new probabilities of all possible states, which represents the queried result.

but does not require more qubits than the Quantum Bayesian Network itself. The inteference step using a SWAP-test, however requires double the qubits (+1) and adds only three gates in depth. Today, neither of both techniques could be executed on an existing quantum device for the given problem. Further, it is expected, that the depth of Grovers algorithm requires a fully fault-tolerant quantum computer. However, in the future, where we have fault-tolerant quantum computers the approach with Grovers algorithm can simulate Bayesian networks with double the size of the SWAP-test approach.

7 Conclusion and outlook

This article focuses on the setup of Quantum Bayesian networks and their application to production engineering, were such networks can be used as a data-driven approach to analyse dependency relations between factors and to estimate the corresponding strength of influences. Quantum computing allows a compact network representation. The discussed potential runtime advantages cannot be proven using the current IBM quantum systems. Even for smaller examples, like the QBN of Fig. 7 without the Grover Search, the circuit depth exceeds the capabilities of current generation quantum computers. The main reason is the multiple controlled *RY*-Gate, which needs to be decomposed into the basis gates of the given quantum computer. Current and future research goals are to implement Quantum Bayesian networks in a more efficient way, analyse further machining operations and to test the developed circuits on a broader range of quantum hardware.

Bibliography

[Bo21] Borujeni, S.E.; Nannapaneni, S.; Nguyen, N.H.; Behrman, E.C.; Steck, J.E.: Quantum circuit representation of Bayesian networks. arXiv:2004.14803, 2021.

[Bu21] Bustillo, A.; Pimenov, D.Y.; Mia, M.; Kapłonek, W.: Machine-learning for automatic prediction of flatness deviation considering the wear of the face mill teeth. Journal of Intelligent Manufacturing, 32:895–912, 2021.

[Ch20] Chapman, P.: Scaling IonQ's Quantum Computers: The Roadmap. IONQ News, 2020.

[DDT20] Deresse, N.C.; Deshpande, V.; Taifa, I.W.R.: Experimental investigation of the effects of process parameters on material removal rate using Taguchi method in external cylindrical grinding operation. Engineering Science and Technology, 23:405–420, 2020.

[GFW21] Gambetta, J.; Faro, I.; Wehden, K.: IBM's roadmap for building an open quantum software ecosystem. IBM Research Blog, 2021.

[Gr96] Grover, Lov K.: A Fast Quantum Mechanical Algorithm for Database Search. arXiv:quant-ph/9605043, November 1996.

[GWG20] Gilliam, Austin; Woerner, Stefan; Gonciulea, Constantin: Grover Adaptive Search for Constrained Polynomial Binary Optimization. arXiv:1912.04088 [quant-ph], August 2020.

[Jo01] Jordan, M.; Kleinberg, J.; Schölkopf, B.; Jensen, F.V.; Nielsen, T.D.: Bayesian Networks and Decision Graphs. Springer New York, New York, NY, 2001.

[KV57] Kienzle, O.; Victor, H.: Spezifische Schnittkräfte bei der Metallbearbeitung. Werkstofftechnik und Maschinenbau, 47:224–225, 1957.

[ORR13] Ozols, M.; Roetteler, M.; Roland, J.: Quantum rejection sampling. ACM Transactions on Computation Theory, 5:1–33, 2013.

[PW09] Pernkopf, F.; Wohlmayr, M.: On Discriminative Parameter Learning of Bayesian Network Classifiers. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD, pp. 212–237, 2009.

[Sa20] Sarma, A.; Chatterjee, R.; Gili, K.; Yu, T.: Quantum unsupervised and supervised learning on superconducting processors. Quantum Information & Computation, 23:541–552, 2020.

[Se21] Selch, M.; Hänel, A.; Frieß, U.; Ihlenfeldt, S.: Quality Monitoring Of Coupled Digital Twins For Multistep Process Chains Using Bayesian Networks. Proceedings of the Conference on Production Systems and Logistics : CPSL 2021, pp. 415–425, 2021.

[SP18] Schuld, M.; Petruccione, F.: Supervised Learning with Quantum Computers. Quantum Science and Technology, 2018.

[YBA18] Yahmadi, R.; Brik, K.; Ammar, F.: Causal tree analysis for quality control of the lead acid battery manufacturing process. International Journal of Energy Research, 42:1738–1759, 2018.

[YC14] Yoder, T.L.and Low, G.H.; Chuang, I.L.: Quantum inference on Bayesian networks. Physical Review A, 89(062315), 2014.