Using quantum computers in control: interval matrix properties

Jan Schneider, Julian Berberich

Abstract—Quantum computing provides a powerful framework for tackling computational problems that are classically intractable. The goal of this paper is to explore the use of quantum computers for solving relevant problems in systems and control theory. In the recent literature, different quantum algorithms have been developed to tackle binary optimization, which plays an important role in various control-theoretic problems. As a prototypical example, we consider the verification of interval matrix properties such as non-singularity and stability on a quantum computer. We present a quantum algorithm solving these problems and we study its performance in simulation. Our results demonstrate that quantum computers provide a promising tool for control whose applicability to further computationally complex problems remains to be explored.

I. INTRODUCTION

Quantum computing has gained increasing attention in recent years due to its ability to solve certain computationally challenging problems more efficiently than classically possible. This includes, for example, integer factorization [1], unstructured search [2], or linear systems of equations [3], but also the simulation of classical [4], [5] and quantum [6], [7, Section 4.7] systems. However, these algorithms cannot be implemented reliably for relevant problem sizes on current noisy intermediate-scale quantum (NISQ) devices [8] due to problems connected to noise and scalability. Variational quantum algorithms (VQAs) are a promising class of quantum algorithms which combine a quantum computer with a classical optimization algorithm [9]. VQAs involve trainable parameters which are optimized iteratively using, e.g., a gradient descent scheme, and, therefore, they are well-suited for NISQ devices due to their adaptation to small and noisy hardware. The quantum approximate optimization algorithm (QAOA) [10] is one of the most popular VQAs and it can be used to solve a specific class of integer programs: quadratic unconstrained binary optimization (QUBO) problems.

Problems with integer variables are relevant in various domains of systems and control theory and, therefore, QAOA is a promising candidate for achieving computational speedups in control. Further quantum algorithms that can tackle computational control problems are listed in [11]. Yet, the usage of quantum computers in control is largely unexplored, except for results on, e.g., MPC with finite input spaces [12] or decentralized control [13]. In this paper, we propose an algorithm for the verification of interval matrix

properties, which is known to be a computationally hard problem, on a quantum computer.

In computational complexity theory, problems can be classified according to the time required for their solution. Problems which can be solved with a polynomial-time algorithm belong to class P and those whose solution can be verified in polynomial time belong to class NP. For NP-hard problems, which are problems to which every problem in NP can be reduced efficiently, there exist no known polynomial-time algorithms on a classical computer. Various control-theoretic problems have been shown to be NP-hard, including static output feedback design or verifying interval matrix properties [14], [15], [16].

In this paper, we introduce an approach for verifying interval matrix properties on a quantum computer. We focus on two main properties: robust non-singularity and robust stability of interval matrices, meaning that all members of the interval matrix are non-singular or stable, respectively. Existing research by [14] shows that the verification of robust non-singularity can be equivalently reformulated as a binary optimization problem. Given that this binary optimization problem is a QUBO problem, it can be tackled using QAOA. The main contribution of this paper is a quantum algorithm based on QAOA which can verify robust non-singularity and robust stability of interval matrices. The applicability is illustrated with numerical examples.

Notation

Let $e = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}^\top$ and define the n-dimensional discrete cube $Q_n = \{-1, 1\}^n$. For each element $z \in Q_n$, the matrix T_z represents a diagonal matrix with the entries of z on its diagonal. Further, for a matrix $A \in \mathbb{R}^{n \times n}$, we define

$$\rho_0(A) = \max\{|\lambda| \,|Ax = \lambda x \text{ for some } x \neq 0, \lambda \in \mathbb{R}\}. \quad (1)$$

The Hermitian conjugate of a matrix $A \in \mathbb{C}^{n \times m}$ is denoted by $A^\dagger$. Finally, I_n denotes an $n \times n$ identity matrix.

II. INTERVAL MATRICES

In this section, we introduce the concept of interval matrices along with the key properties that are studied in the present paper. An interval matrix is defined as

$$A^I = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{n \times m} | \underline{A} \leq A \leq \overline{A}\} \subseteq \mathbb{R}^{n \times m}, \quad (2)$$

where $\overline{A}$ and $\underline{A}$ represent an upper and lower bound matrix and the inequality is interpreted element-wise. Further, we define the center of an interval matrix as

$$A_m = \frac{1}{2}(\overline{A} + \underline{A}). \quad (3)$$

This allows to reformulate the lower and upper bound matrices as $\overline{A} = A_m + \Delta_d$ and $\underline{A} = A_m - \Delta_d$ with $\Delta_d = \frac{1}{2}(\overline{A} - \underline{A})$.

A. Non-singularity of interval matrices

In this paper, we study non-singularity of interval matrices based on the radius of non-singularity. The definition of a non-singular interval matrix is given in the following.

Definition 1. *An interval matrix A^I is non-singular if all matrices $A \in A^I$ are non-singular.*

The radius of non-singularity measures the distance of a matrix A to the closest singular matrix according to an a priori fixed shifting matrix Δ and is defined as

$$d(A, \Delta) = \min_{\varepsilon \geq 0} \quad \varepsilon \tag{4}$$
$$\text{s.t.} \quad A - \varepsilon\Delta \leq A' \leq A + \varepsilon\Delta$$
$$\text{for some singular } A' \in \mathbb{R}^{n \times n}.$$

This problem minimizes ε such that the interval matrix $[A - \varepsilon\Delta, A + \varepsilon\Delta]$ is singular, i.e., contains a singular matrix $A' \in [A - \varepsilon\Delta, A + \varepsilon\Delta]$. Given the optimal value ε^* of problem (4), a non-singular interval matrix can be defined accordingly as

$$[A - \varepsilon\Delta, A + \varepsilon\Delta] \tag{5}$$

for any $\varepsilon < \varepsilon^*$. The radius of non-singularity is relevant, e.g., for sensor processing in the presence of noise [14] or for sensitivity analysis of linear systems [17], and it has close connections to various concepts including the structured singular value which is a powerful tool in robust stability analysis [18]. Further, as we show below, it can be used to study robust stability properties of uncertain linear systems.

As an instructive example, let us consider the matrices

$$A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \Delta = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{6}$$

The boundaries of the corresponding interval matrix considered in (4) are given by

$$A \pm \varepsilon\Delta = \begin{bmatrix} 1 \pm \varepsilon & -1 \pm \varepsilon \\ \pm\varepsilon & 1 \pm \varepsilon \end{bmatrix}. \tag{7}$$

The optimal solution of (4) is given by $\varepsilon^* = \frac{1}{3}$ with the associated singular matrix

$$A' = A - \frac{1}{3}\Delta = \begin{bmatrix} \frac{2}{3} & -\frac{4}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{bmatrix}. \tag{8}$$

As shown in [14], the radius of non-singularity of a given, non-singular matrix A and a matrix Δ, can be reformulated as a combinatorial optimization problem:

$$d(A, \Delta) = \frac{1}{\max\{\rho_0(A^{-1}T_y\Delta T_z)|y, z \in Q_n\}}. \tag{9}$$

The optimization problem appearing in the denominator is the central problem considered in this paper. The proof of NP-hardness is given in [14] and is sketched briefly in the

following. The maximization in (9) can be simplified for the specific choice $\Delta = ee^\top$, which allows to reformulate (9) as

$$d(A) = \frac{1}{r(A^{-1})} \tag{10}$$

with

$$r(M) = \max\{z^\top My|z, y \in Q_n\}. \tag{11}$$

It is shown in [14] that the *maximum-cut problem* can be reduced to the optimization problem (11). Since the maximum-cut problem is NP-hard, it follows that computing the radius of non-singularity is NP-hard as well.

In this work, we derive a combinatorial optimization problem similar to (10) but for a more general class of matrices Δ than considered in [14]. To be precise, the condition $\Delta = ee^\top$ is relaxed to Δ having rank 1, allowing for more versatile and practically relevant problem considerations.

B. Stability of interval matrices

Beyond non-singularity, we also address the problem of stability verification of interval matrices.

Definition 2. *An interval matrix A^I is stable if all matrices $A \in A^I$ are stable, i.e., $\mathrm{Re}(\lambda_i(A)) < 0$ for all eigenvalues $\lambda_i(A)$ of A.*

Stability of an interval matrix A^I implies robust stability of the uncertain system

$$\dot{x} = Ax \tag{12}$$

with $A \in A^I$, which is an important problem in robust control [19]. In [20], it is shown that, just like non-singularity, verifying stability of interval matrices is NP-hard as well. In the following result, we exploit that stability and non-singularity are closely related due to the continuity of eigenvalues in the matrix entries. For simplicity, we focus on symmetric interval matrices. Considering general interval matrices is an interesting issue for future research.

Definition 3. *For an interval matrix A^I, the symmetric interval matrix A_{sym}^I is defined as*

$$A_{\mathrm{sym}}^I = \{A \in A^I \mid A = A^\top\}. \tag{13}$$

Stability verification of a symmetric interval matrix is relevant in scenarios, where not only uncertainty bounds specified by A^I are available but also additional structural knowledge on symmetry of the dynamics. The following result shows that stability of symmetric interval matrices can be verified based on a non-singularity test.

Proposition 4. *Suppose A_{sym}^I is a non-singular symmetric interval matrix and there exists a stable $A \in A_{\mathrm{sym}}^I$. Then, A_{sym}^I is stable.*

Proof. If $A \in A_{\mathrm{sym}}^I$ is stable, then all its eigenvalues have real value less than 0. Note that all eigenvalues of the matrices in A_{sym}^I are real due to symmetry. Since eigenvalues are continuous functions in the matrix entries, the non-singularity of A_{sym}^I implies that all $A' \in A_{\mathrm{sym}}^I$ are stable. $\square$

Motivated by Proposition 4, we focus on deriving a quantum algorithm for verifying robust non-singularity, which then also allows to verify robust stability of a symmetric interval matrix by testing stability of one of its elements. Note that the eigenvalues of a single matrix can be computed efficiently on a classical computer, which is the only additional step for the stability check.

III. QUANTUM ALGORITHM

In this section, we propose the quantum algorithm for verifying the interval matrix properties introduced above. First, in Section III-A, we present basic elements of quantum computing that are required to state and implement our algorithm. Next, in Section III-B, the key underlying algorithm QAOA is explained in more detail followed by the construction of the problem Hamiltonian in Section III-C. Finally, the overall algorithm is stated in Section III-D.

A. Quantum Basics

In the following, we present required basics of quantum computing. For further details, we refer to the tutorial [11], which introduces quantum computing from the perspective of control, as well as to the textbook [7]. Qubits (short for "quantum bits") are the basic components of a quantum computer, comparable to bits in classical computing. Mathematically, a qubit is written as

$$|\psi\rangle = \begin{bmatrix} z_0 \\ z_1 \end{bmatrix} \tag{14}$$

for some $z_0, z_1 \in \mathbb{C}$ with $|z_0|^2 + |z_1|^2 = 1$. We use the standard Dirac notation $|\psi\rangle$ for denoting quantum states and $\langle\psi| = |\psi\rangle^\dagger$ for their Hermitian conjugate. The precise value of $|\psi\rangle$ is not accessible. Instead, a measurement can only reveal that the system is in one of the two basis states

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{15}$$

where the probabilities for either outcome are $|z_0|^2$ and $|z_1|^2$, respectively. A quantum state consisting of n qubits lives in $\mathbb{C}^{2^n}$.

The second fundamental component of a quantum computer are quantum gates. These are used to manipulate the qubits and they are represented by unitary matrices U, i.e., they satisfy $U^\dagger U = I$. The application of a quantum gate U to an input state $|\psi_{\text{in}}\rangle$ is defined by multiplication, i.e., the output state is

$$|\psi_{\text{out}}\rangle = U |\psi_{\text{in}}\rangle. \tag{16}$$

The *Pauli gates* X, Y, and Z are popular examples of quantum gates and they are defined as

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{17}$$

For example, applying the X gate to a qubit swaps the amplitudes of $|0\rangle$ and $|1\rangle$ as can be seen by

$$X |0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |1\rangle \tag{18}$$
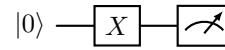

Fig. 1. Quantum circuit implementing the operation (18) followed by a measurement.

and similarly $X |1\rangle = |0\rangle$. The final part of a quantum algorithm is the measurement. In Figure 1, a simple example of a quantum circuit is given which implements the operation (18) with subsequent measurement of the qubit.

As mentioned above, qubits cannot be measured directly, but a measurement can only return one of a finite number of possible outcomes, where the corresponding probabilities are determined by the probability amplitudes z_0, z_1 of the qubit, cf. (14). To be more precise, measurements are always taken w.r.t. an *observable* $\mathcal{M} = \mathcal{M}^\dagger$. Repeated measurements of a state $|\psi\rangle$ w.r.t. the observable $\mathcal{M}$ allow to retrieve the expectation value

$$\langle\psi|\mathcal{M}|\psi\rangle = \psi^\dagger \mathcal{M}\psi. \tag{19}$$

For many quantum algorithms, in particular VQAs, this expectation value is the actual output of the algorithm.

B. Quantum approximate optimization algorithm

VQAs are a class of quantum algorithms combining an optimization procedure on a classical computer with a quantum algorithm [9]. This allows the algorithm to adapt to noise on the quantum computer and, thereby, provide possibly more reliable results. QAOA [10] is an important example of VQAs. It is tailored to solving combinatorial optimization problems and involves alternating between a classical optimization routine to determine parameters of quantum gates and a quantum evolution via the resulting parametrized circuit.

In the following, we provide a brief introduction to QAOA. QAOA can be used to solve problems of the form

$$\max_{x \in \{0,1\}^n} \quad C(x) \tag{20}$$

for a cost function $C : \{0,1\}^n \to \mathbb{R}$. The algorithm consists of repeated applications of parametrized unitaries to an initial quantum state $|\psi_0\rangle = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^\top$. The two unitaries applied to this state are the *mixing unitary* $U_{\text{B}}(\beta_j) = e^{-i\beta_j H_{\text{B}}}$ with mixing Hamiltonian $H_{\text{B}} = H_{\text{B}}^\dagger$ and the *problem unitary* $U_{\text{P}}(\gamma_j) = e^{-i\gamma_j H_{\text{P}}}$ with problem Hamiltonian $H_{\text{P}} = H_{\text{P}}^\dagger$. The parameter vectors $\beta, \gamma \in \mathbb{R}^p$ are optimization variables for the p layers of the quantum circuit. These two unitaries are applied to the initial state $|\psi_0\rangle$ in an alternating fashion

$$U(\beta_p, \gamma_p) \cdots U(\beta_1, \gamma_1) \tag{21}$$

with $U(\beta_j, \gamma_j) = U_{\text{B}}(\beta_j)U_{\text{P}}(\gamma_j)$. The number of alternations p in (21) is problem-specific but typically chosen small to enable implementations on current quantum hardware. Figure 2 shows the 2-qubit quantum circuit consisting of a measurement of the state

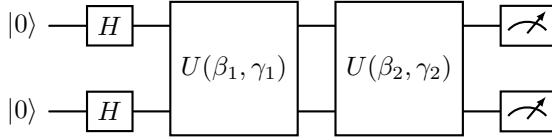$$U(\beta_2, \gamma_2)U(\beta_1, \gamma_1) |\psi_0\rangle. \tag{22}$$

Fig. 2. Circuit representation of QAOA for $n = 2$ qubits and $p = 2$ layers.

Fig. 3. Basic scheme of QAOA as an iterative optimization containing classical parameter updates and executions of parametrized quantum circuits.

Here, H denotes the *Hadamard* gate defined as

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{23}$$

which generates the input state due to

$$|\psi_0\rangle = (H \otimes H)(|0\rangle \otimes |0\rangle) \tag{24}$$

with the Kronecker product $\otimes$.

Let us now discuss how the mixing and problem Hamiltonians H_B and H_P are chosen. The problem Hamiltonian encodes the cost function $C(x)$ into the quantum algorithm, see Section III-C for the construction procedure. On the other hand, the mixing Hamiltonian is used for entangling the solution amplitudes and expanding the solution space to create a diverse set of candidate solutions. The mixing Hamiltonian is commonly chosen independently of the cost function $C(x)$ as

$$H_B = \sum_{j=1}^{n} X_j, \tag{25}$$

where we use the standard notation

$$X_j = \underbrace{I_2 \otimes \cdots \otimes I_2}_{j-1 \text{ times}} \otimes X \otimes \underbrace{I_2 \otimes \cdots \otimes I_2}_{n-j \text{ times}}. \tag{26}$$

Finally, the expectation value

$$\langle \psi(\beta, \gamma)|H_P|\psi(\beta, \gamma)\rangle = \psi(\beta, \gamma)^\dagger H_P \psi(\beta, \gamma) \tag{27}$$

of the parametrized state $|\psi(\beta, \gamma)\rangle$ is determined based on repeated measurements, compare (19). Based on this value, a classical computer determines a new set of parameters γ and β maximizing (27) using some optimization scheme, e.g., gradient descent. This procedure is then repeated until convergence. Figure 3 summarizes the basic scheme of QAOA as an iteration between classical optimization and execution of the quantum algorithm (27).

After converging to a set of parameters β^*, γ^*, the candidate solution for (20) is obtained by performing repeated measurements of the resulting quantum state

$$\begin{aligned}|\psi(\beta^*, \gamma^*)\rangle \\ = a_1 |0\ldots00\rangle + a_2 |0\ldots01\rangle + \cdots + a_{2^n} |1\ldots11\rangle,\end{aligned} \tag{28}$$

where we use the standard notation for *computational basis states*, e.g.,

$$|0...011\rangle = |0\rangle \otimes \cdots \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle. \tag{29}$$

A measurement of $|\psi(\beta^*, \gamma^*)\rangle$ returns the j-th computational basis state with probability $|a_j|^2$. The computational basis state with the highest pro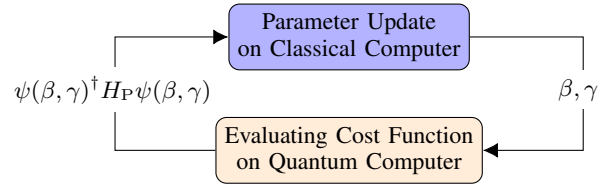bability (determined empirically by counting occurrences in repeated measurements) is then employed as candidate solution for the original binary optimization problem (20).

C. Formulating the problem Hamiltonian

In the following, we show how the problem Hamiltonian H_P can be chosen in order to compute the radius of non-singularity via QAOA. Recall the combinatorial optimization problem introduced in (9), i.e.,

$$\max \bar{C}(y, z) = \max\{\rho_0(A^{-1}T_y \Delta T_z)|y, z \in Q_n\}. \tag{30}$$

In order to apply QAOA, we now transform (30) into a binary optimization problem via a variable transformation from ± 1 to $0/1$. More precisely, we introduce the binary variable $x \in \{0,1\}^n$ according to $x_i = \frac{1-z_i}{2}$ and $x_{i+n} = \frac{1-y_i}{2}$, resulting in

$$(-1)^{x_i} = z_i, \quad (-1)^{x_{i+n}} = y_i. \tag{31}$$

In order to construct a suitable H_P, the relation

$$C(x) |x\rangle = H_P |x\rangle \tag{32}$$

needs to hold for any $x \in \{0,1\}^n$. Here, $C(x)$ is the cost in the new parametrization such that $C(x) = \bar{C}(y, z)$ for all $x \in \{0,1\}^n$ and y, z satisfying (31). Indeed, multiplying (32) by $\langle x|$ from the left-hand side and using unit norm of $|x\rangle$ yields $C(x) = \langle x|H_P|x\rangle$, which is why QAOA aims at maximizing $\langle x|H_P|x\rangle$, compare (27).

The following result provides a possible choice of H_P satisfying (32).

Theorem 5. *Suppose* $\text{rank}(\Delta) = 1$ *and define the problem Hamiltonian*

$$H_P = \sum_{i,j=1}^{n} v_i \tilde{a}_{ij} \delta_j Z_i Z_{n+j}$$

with $(\tilde{a}_{ij}) = A^{-1}$, Z_i *as in (26) for* $X = Z$, *and* $\delta, v \in \mathbb{R}^n$ *such that* $\Delta = \delta v^\top$. *Then, (32) holds.*

Proof. First, (30) is reformulated using that $\Delta = \delta v^\top$, i.e., $\text{rank}(\Delta) = 1$. To this end, if λ is a non-zero, real eigenvalue of $A^{-1}T_y \Delta T_z$, it holds that

$$A^{-1}T_y \Delta T_z w = A^{-1}T_y \delta v^\top T_z w = \lambda w. \tag{33}$$

Since $\lambda \neq 0$, it follows that $v^\top T_z w \neq 0$. Together with (33), this implies

$$A^{-1}T_y \delta = \lambda w(v^\top T_z w)^{-1}. \tag{34}$$

Left-multiplying by $v^\top T_z$ yields

$$v^\top T_z A^{-1} T_y \delta = \lambda. \tag{35}$$

Subsequently, this results in

$$\max \bar{C}(y,z) = \max\{|v^\top T_z A^{-1} T_y \delta| \,|\, y,z \in Q_n\} \tag{36}$$
$$= \max\{v^\top T_z A^{-1} T_y \delta | y,z \in Q_n\}.$$

The absolute value can be neglected due to the (bi-)linear cost function and the symmetry of Q_n. Substituting (31) into $\bar{C}(y,z)$, we obtain

$$C(x) = v^\top T_{x_z} A^{-1} T_{y_z} \delta \tag{37}$$

$$= \begin{bmatrix} x_{z,1} v_1 \\ \vdots \\ x_{z,n} v_n \end{bmatrix}^\top A^{-1} \begin{bmatrix} x_{y,1} \delta_1 \\ \vdots \\ x_{y,n} \delta_n \end{bmatrix}$$

$$= \sum_{i,j=1}^{n} v_i \tilde{a}_{ij} \delta_j (-1)^{x_i} (-1)^{x_{n+j}}$$

with

$$x_z = \begin{bmatrix} (-1)^{x_1} & \ldots & (-1)^{x_n} \end{bmatrix}^\top \tag{38}$$
$$x_y = \begin{bmatrix} (-1)^{x_{n+1}} & \ldots & (-1)^{x_{2n}} \end{bmatrix}^\top.$$

Note that the Pauli-Z gate satisfies

$$Z |x\rangle = (-1)^x |x\rangle, x \in \{0,1\}. \tag{39}$$

Applied to (37), this results in

$$C(x) |x\rangle = \sum_{i,j=1}^{n} v_i \tilde{a}_{ij} \delta_j Z_i Z_{n+j} |x_0, \ldots, x_{2n-1}\rangle \tag{40}$$
$$= H_{\mathrm{P}} |x\rangle. \qquad \square$$

The proof of Theorem 5 consists of two parts: 1) reformulation of the optimization problem (30) as a QUBO and 2) reformulation of the QUBO as a VQA by choosing a suitable problem Hamiltonian H_{P}. The step 1) is inspired by [14] but faces the additional technical challenge of considering a general rank-1 matrix Δ rather than $\Delta = ee^\top$. Further, the step 2) follows the derivation in [21], adapted to the specific binary optimization problem considered in the present paper.

D. The Quantum Algorithm

Algorithm 6 summarizes the overall quantum algorithm verifying robust non-singularity of a given interval matrix A^I. It can also be used to verify stability of a symmetric interval matrix by using Proposition 4.

IV. IMPLEMENTATION

In order to examine the performance of the proposed algorithm, we simulate the algorithm using the Pennylane toolbox [22] for Python. The source code for the implementation is publicly accessible on `https://github.com/JanKyb/Radius-Of-NonSingularity-using-QAOA`.

To verify the performance, we consider two examples. For the first example, the matrices of (6) are reconsidered.

Algorithm 6. Verifying non-singularity of an interval matrix

Input: $A^I = [A_m - \varepsilon\Delta, A_m + \varepsilon\Delta]$ and $\Delta = (\delta_{ij})$ with $\delta_{ij} \geq 0$ and $\mathrm{rank}(\Delta) = 1$.
1) Compute A_m^{-1}
2) Construct problem Hamiltonian H_{P} (cf. Theorem (5))
3) Run QAOA to compute $d(A_m, \Delta)$ (cf. Section III-B and [10] for details)
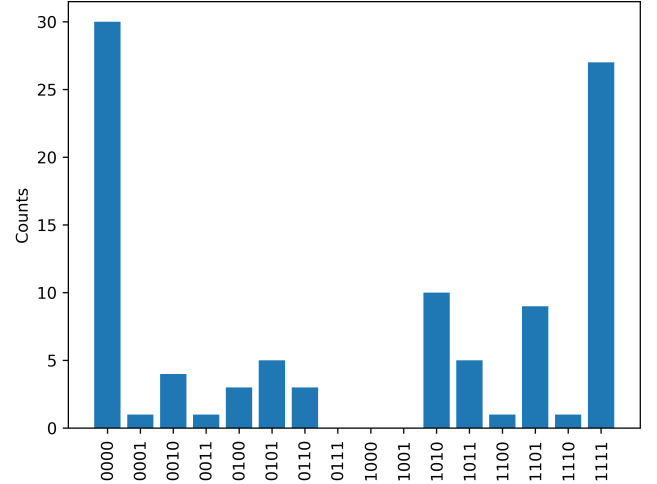4) If $d(A_m, \Delta) > \varepsilon$, then A^I is non-singular

---

Fig. 4. Counts per string for the output of the proposed algorithm when applied to the first example.

Calculating the radius of non-singularity with the proposed quantum algorithm leads to the results in Figure 4. The counts shown in the figure represent an approximation of the probabilities $|a_j|^2$ corresponding to the quantum state (28) obtained via the proposed algorithm, followed by an additional conversion step according to (31).

It can be seen, that the first string 0000 yields the solution with the (empirically) highest probability, i.e., the maximum amount of counts. Transforming this string back to the initial coordinates based on (31), the corresponding candidate solution of (30) is given by $z = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ and $y = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$. This yields the result $d(A, \Delta) = \frac{1}{3}$, which is indeed the correct solution, compare Section II. Note that also the solution string 1111 has a comparably large amount of counts. Evaluating the cost function, 1111 also yields the same optimal result as 0000, which is due to the symmetry in the variable transformation (31) and the cost function in (30). Moreover, several further strings corresponding to a suboptimal solution also have a non-trivial amount of counts. This characteristic is representative for QAOA being a quantum algorithm with an inherently probabilistic output.

As a second and more realistic example, we study robust stability of an RL circuit from [23], which is a prime example

binary strings	counts
000111	48
111000	33
111001	5
000110	2
010011	2

TABLE I

Counts per String for the output of the proposed algorithm when applied to the second example

of a symmetric system. The corresponding system matrix is

$$A = \begin{bmatrix} -2 & 2 & 0 \\ 2 & -5 & 3 \\ 0 & 3 & -7 \end{bmatrix}. \quad (41)$$

To analyze robust stability of this system subject to additional uncertainty, we are interested in finding a possibly large value ε such that all symmetric matrices in $[A - \varepsilon\Delta, A + \varepsilon\Delta]$ are stable, where we consider $\Delta = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top$. In the following, this analysis will be carried out by computing the radius of non-singularity $d(A, \Delta)$ via the proposed quantum algorithm.

An application of Algorithm 6 leads to a distribution of counts analogous to Figure 4. Due to the increased dimension, we do not depict all possible solutions but only list the bit strings with the most counts in Table I.

The solution with the highest number of counts is 000111. To verify that this is indeed an optimal solution, we use (31) to transform the optimal string 000111 into the initial coordinates as $z = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top$ and $y = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}^\top$. Plugging this candidate into the cost of (30) yields 4.0833. Thus, we have $d(A, \Delta) = \frac{1}{4.0833}$ and, indeed, $A - \frac{1}{4.0833}\Delta$ is (approximately, i.e., modulo numerical inaccuracies) singular. Using that A is stable, Proposition 4 implies that all symmetric matrices in $[A - \varepsilon\Delta, A + \varepsilon\Delta]$ for any $\varepsilon < 4.0833$ are stable.

Finally, note that the candidate 111000 only has (compared to the other candidates) slightly less counts than the best solution 000111 since both candidates are, in fact, equivalent. This follows again from the symmetry in the variable transformation (31) and the cost function in (30).

V. Conclusion

Quantum computing is a rapidly advancing technology that promises to solve certain computational problems faster than classically possible. In this paper, we presented a quantum algorithm for verifying non-singularity and stability of interval matrices, which are relevant problems, e.g., in robust stability analysis. The proposed algorithm relies on QAOA which is a popular recent quantum algorithm addressing combinatorial optimization. Extending our results to stability analysis of general (not symmetric) interval matrices as well as the implementation on a real quantum computer are interesting issues for future research. Moreover, given the high relevance of combinatorial optimization problems in various domains of control, applying QAOA to solve computationally complex problems in control poses another promising future research direction. Beyond combinatorial optimization, further computational problems appearing in control may as well be amenable to quantum computing, see [11] for an overview.

References

[1] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th ACM Symposium on the Theory of Computing*, 1996, pp. 212–219.

[3] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, p. 150502, 2009.

[4] D. Giannakis, A. Ourmazd, P. Pfeffer, J. Schumacher, and J. Slawinska, "Embedding classical dynamics in a quantum computer," *Physical Review A*, vol. 105, p. 052404, 2022.

[5] M. A. Schalkers and M. Möllers, "Efficient and fail-safe collisionless quantum Boltzmann method," *arXiv:2211.14269*, 2022.

[6] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, p. 467, 1982.

[7] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. Cambridge University Press, New York, NY, USA, 2011.

[8] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[9] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, pp. 625–644, 2021.

[10] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv:1411.4028*, 2014.

[11] J. Berberich and D. Fink, "Quantum computing through the lens of control: a tutorial introduction," *arXiv:2310.12571*, 2023.

[12] D. Inoue and H. Yoshida, "Model predictive control for finite input systems using the D-wave quantum annealer," *arXiv:2001.01400*, 2020.

[13] S. A. Deshpande and A. A. Kulkarni, "The quantum advantage in decentralized control," *arXiv:2207.12075*, 2022.

[14] S. Poljak and J. Rohn, "Checking robust nonsingularity is NP-hard," *Mathematics of Control, Signals, and Systems*, vol. 6, pp. 1–9, 1993.

[15] A. Nemirovskii, "Several NP-hard problems arising in robust stability analysis," *Math. Control Signals Systems*, vol. 6, pp. 99–105, 1993.

[16] V. Blondel and J. N. Tsitsiklis, "NP-hardness of some linear control design problems," *SIAM J. Control Optim.*, vol. 35, no. 6, pp. 2118–2127, 1997.

[17] A. Deif, *Sensitivity analysis in linear systems*. Springer-Verlag, Berlin, 1986.

[18] J. C. Doyle, "Analysis of feedback systems with structured uncertainties," in *Proc. IEEE*, 1982, pp. 242–250.

[19] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1996.

[20] J. Rohn, "Checking positive definiteness or stability of symmetric interval matrices is NP-hard," *Comment. Math. Univ. Carolin.*, vol. 35, no. 4, pp. 795–797, 1994.

[21] S. Hadfield, "On the representation of boolean and real functions as hamiltonians for quantum computing," *ACM Transactions on Quantum Computing*, vol. 2, no. 4, p. 18, 2021.

[22] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv:1811.04968*, 2018.

[23] M. Meisami-Azad, J. Mohammadpour, and K. M. Grigoriadis, "Dissipative analysis and control of state-space symmetric systems," in *Proc. American Control Conf. (ACC)*, 2008, pp. 413–418.