DIGITALCOMMONS
@WAYNESTATE

Wayne State University

Chemical Engineering and Materials Science Faculty Research Publications

Chemical Engineering and Materials Science

6-17-2022

Quantum Computing and Resilient Design Perspectives for Cybersecurity of Feedback Systems

Keshav Kasturi Rangan
Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI, keshav@wayne.edu

Jihan Abou Halloun
Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI

Henrique Oyama
Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI, hcoyama@wayne.edu

Samantha Cherney
Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI

Ilham Azali Assoumani
Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI

Follow this and additional works at: https://digitalcommons.wayne.edu/cems_eng_frp

e next page for additional authors

Part of the Controls and Control Theory Commons, Information Security Commons, and the Process Control and Systems Commons

Recommended Citation

Rangan, K. K., J. Abou Halloun, H. Oyama, S. Cherney, I. Azali Assoumani, N. Jairazbhoy, H. Durand, and S. K. Ng, "Quantum Computing and Resilient Design Perspectives for Cybersecurity of Feedback Systems," IFAC-PapersOnLine (Proceedings of the 13th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2022: Busan, Republic of Korea, 14–17 June 2022), 55(7), 703-708. https://doi.org/10.1016/j.ifacol.2022.07.526

This Conference Proceeding is brought to you for free and open access by the Chemical Engineering and Materials Science at DigitalCommons@WayneState. It has been accepted for inclusion in Chemical Engineering and Materials Science Faculty Research Publications by an authorized administrator of DigitalCommons@WayneState.

Authors

Keshav Kasturi Rangan, Jihan Abou Halloun, Henrique Oyama, Samantha Cherney, Ilham Azali Assoumani, Nazir Jairazbhoy, Helen Durand, and Simon Ka Ng

This conference proceeding is available at DigitalCommons@WayneState: https://digitalcommons.wayne.edu/cems_eng_frp/17

Quantum Computing and Resilient Design Perspectives for Cybersecurity of Feedback Systems

Keshav Kasturi Rangan * **Jihan Abou Halloun** *
Henrique Oyama * **Samantha Cherney** *
Ilham Azali Assoumani * **Nazir Jairazbhoy** * **Helen Durand** *
Simon Ka Ng *

* *Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI 48202 USA.*

Abstract: Cybersecurity of control systems is an important issue in next-generation manufacturing that can impact both operational objectives (safety and performance) as well as process designs (via hazard analysis). Cyberattacks differ from faults in that they can be coordinated efforts to exploit system vulnerabilities to create otherwise unlikely hazard scenarios. Because coordination and targeted process manipulation can be characteristics of attacks, some of the tactics previously analyzed in our group from a control system cybersecurity perspective have incorporated randomness to attempt to thwart attacks. The underlying assumption for the generation of this randomness has been that it can be achieved on a classical computer; however, quantum computers can also create random behavior in the results of computations. This work explores how errors in quantum hardware that can create non-deterministic outputs from quantum computers interact with control system cybersecurity. These studies serve as a reminder of the need to incorporate cybersecurity considerations at the process design stage.

Keywords: quantum computing, control, resilience, cybersecurity

1 INTRODUCTION

Attacks on process control systems can provide a means for an attacker to use a computing system to impact a safety-critical physical system. Prior work in our group has attempted to thwart attacks via control and detection implementation strategies that force attacks to show themselves before a safety issue can occur (e.g., Oyama et al. (2022)). Though randomness in control action computation has been considered in strategies discussed in Durand (2018); Oyama and Durand (2020), it has not in these works been able to guarantee that attacks cannot create safety hazards. However, these control designs have been implemented on classical computers. Recently, quantum computing has been applied to problems in process systems engineering such as optimization and machine learning (Ajagekar et al. (2020); Ajagekar and You (2019)). Quantum computers can also introduce randomness to the results of the computations which they execute. One way that they can do this is through errors ("noise") in the computations on present-day devices. This work will therefore investigate relationships between randomness introduced by noise in a quantum computation and stabilization of a closed-loop state trajectory where the control actions are computed by a quantum simulator. This will be used to extend the understanding of the role of randomness in cyberattack detection and handling strategies; however, as in our prior work, we will not see a benefit from the use of the randomness for attempting to thwart an attack. This serves as a reminder of the complexity of attacks and the need to consider them at the design stage. We close with remarks regarding how cybersecurity considerations might impact evaluation of results from a steady-state simulator during the design process of a geothermal energy process.

2 Motivation: Cyberattacks are Distinct from Disturbances, Faults, and Measurement Noise

In this section, we motivate the need for studying control system cyberattacks using a continuous stirred tank reactor (CSTR) where a second-order exothermic reaction $A \rightarrow B$ occurs with the following dynamics:

$$\dot{C}_A = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{-\frac{E}{R_g T}} C_A^2 + b_1 + w_1 \quad (1)$$

$$\dot{T} = \frac{F}{V}(T_0 - T) - \frac{\Delta H k_0}{\rho_L C_p} e^{-\frac{E}{R_g T}} C_A^2 + \frac{Q}{\rho_L C_p V} + b_2 + w_2 \quad (2)$$

where $b_1 = 5$ and $b_2 = 50$ represent a bias in the dynamics compared to a standard mass and energy balance for a CSTR, which might be attributed to unmodeled dynamics or faults. The reactant concentration C_A and temperature T in the reactor are the states, and C_{A0} and Q (the inlet concentration of the reactant and the heat rate, respectively) are the inputs. The parameters of the model are taken from Alanqar et al. (2015). w_1 and w_2 are bounded disturbances with zero mean and standard deviations of 2 kmol/m^3 · h and 10 K/h and bounds of 1 kmol/m^3 · h and 5 K/h, respectively. Measurement noise was also considered with zero mean and standard deviations of 0.005 kmol/m^3 and 0.1 K, and bounds of 0.05 kmol/m^3 and 1 K, respectively. These were implemented using the

function "normal_distribution" in C++ with the seed set to one greater than the sampling period number (the first sampling period corresponds to the time from t_0 to t_1).

It is assumed that the fact that the bias terms 5 and 50 exist on the right-hand sides of Eqs. 1-2 is not known, and that therefore data is gathered by operating the system under an EMPC that assumes that the process model does not have these biases (or disturbances/noise) to observe relationships between states and inputs. The EMPC used in this case was implemented in Ipopt (Wächter and Biegler (2006)) with ADOL-C (Walther and Griewank (2009)) using code for integrating IPOPT and ADOL-C from Walther (2010) and has the following form:

$$\min_{C_{A0},Q\in S(\Delta),s} \int_{t_k}^{t_{k+N}} \left[-k_0 e^{-\frac{E}{R_g \tilde{T}(\tau)}} \tilde{C}_A(\tau)^2 + 10^8 s^2\right] d\tau \quad (3a)$$

$$\text{s.t. Eq. } 1, b_1 = 0, b_2 = 0, w_1 = 0, w_2 = 0 \quad (3b)$$

$$\tilde{C}_A(t_k) = \bar{C}_A(t_k), \tilde{T}(t_k) = \bar{T}(t_k) \quad (3c)$$

$$\tilde{T}(t) \leq 450 \text{ K} + s, \ \forall t \in [t_k, t_{k+N}) \quad (3d)$$

$$0.5 \leq C_{A0} \leq 7.5 \text{ kmol/m}^3, \ \forall t \in [t_k, t_{k+N}) \quad (3e)$$

$$|Q| \leq 5 \times 10^5 \text{ kJ/h}, \ \forall t \in [t_k, t_{k+N}) \quad (3f)$$

$$0 \leq s \leq 5 \quad (3g)$$

where $\bar{C}_A(t_k)$ and $\bar{T}(t_k)$ represent the state measurement (subject to measurement noise) at t_k, and s represents a slack variable to handle the plant/model mismatch. Eq. 3a represents an economics-based objective function, Eq. 3d bounds the temperature in the reactor (and is enforced at the end of every integration step of 10^{-4} h used within the EMPC with the explicit Euler numerical integration method), and Eqs. 3e-3f represent input constraints. The process of Eqs. 1-2 with disturbances added is integrated using the explicit Euler numerical integration method with an integration step size of 10^{-5} h. Data is collected for 10 sampling periods of length 0.01 h, where $N = 10$. The resulting data is then fed to a separate optimization problem (in the spirit of moving horizon estimation (Alessandri et al. (2010))) to estimate the bias terms as follows:

$$\min_{C_A(t_0),\tilde{T}(t_0);b_1,b_2} \int_{t_k}^{t_{k+N}} [10^4(\tilde{C}_A(\tau) - C_A(\tau))^2 + (\tilde{T}(\tau) - T(\tau))^2] \, d\tau \quad (4a)$$

$$\text{s.t. Eq. } 1, w_1 = 0, w_2 = 0 \quad (4b)$$

$$\tilde{C}_A(t_0) = C_A(t_0), \tilde{T}(t_0) = T(t_0) \quad (4c)$$

$$0 \leq C_A(t_0) \leq 5 \text{ kmol/m}^3, \ \forall t \in [t_k, t_{k+N}) \quad (4d)$$

$$0 \leq T(t_0) \leq 500 \text{ K}, \ \forall t \in [t_k, t_{k+N}) \quad (4e)$$

$$|b_1| \leq 500 \text{ kmol/m}^3 \cdot \text{h}, \ \forall t \in [t_k, t_{k+N}) \quad (4f)$$

$$|b_2| \leq 10000 \text{ K/h}, \ \forall t \in [t_k, t_{k+N}) \quad (4g)$$

where b_1 and b_2 are decision variables representing the bias terms on the right-hand side of Eqs. 1-2. Eq. 4a reflects that the error between the measurements obtained from inputs computed using the first optimization problem (Eq. 3) and the predictions made using the models in Eq. 4b is minimized in this equation, using the same inputs recorded from solving the optimization problem of Eq. 3. The initial guess provided to the optimization problem is

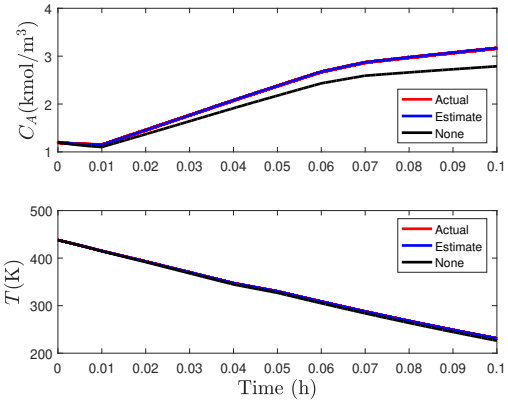

Fig. 1. State trajectories under inputs from Eq. 4 with different process models. The data from the "Actual" model may differ from that used in the identification of the biases due to the random number generator computing different noise profiles.

$C_{A0}(t_0) = 1.1$ kmol/m^3, $T(t_0) = 435$ K, $b_1 = 0.1$, and $b_2 = 5$. The solution to this optimization problem from Ipopt using the data from t_0 until right before t_{10} gives an initial state estimate of $C_A(t_0) = 1.199$ kmol/m^3 and $T(t_0) = 437.982$ K, where $b_1 = 5.065$ and $b_2 = 50.934$. To evaluate the performance of the model incorporating the biases, we compare the state trajectories from the initial condition under the same inputs when both the actual system model (including disturbances and noise) is used ("Actual" in Fig. 1), when the model without the biases or disturbances or noise is used ("None"), and when the model with the estimated biases but no disturbances or noise is used ("Estimate"). These indicate that the model can be significantly improved by accounting for the fault/plant-model mismatch. It would be expected that the more accurate model would provide better control in an MPC. The difference between a fault and a disturbance can be highlighted in this discussion; specifically, even if a cyberattack (e.g., on the actuators) was to affect the right-hand side of Eqs. 1-2 in the same manner as the "faults" currently accounted for in that equation, the cyberattack could simultaneously include falsification of the measurement data (for example, data corresponding to the case without noise or disturbances might be provided instead). This would prevent Eq. 4 from computing adequate values of b_1 and b_2 so that the system is not able to update the controller to account for the attacks on the actuators. This constitutes the fundamental difference between cyberattacks and faults: while a fault and cyberattack may affect the system dynamics in similar ways, cyberattacks involve a coordination to prevent the system from compensating for issues as it would in the case of a fault. This makes handling cyberattacks particularly challenging, as it requires not only strategies for handling them once they are detected, but also for revealing that they exist.

3 Control System Resilience and Quantum Computing-Implemented Control

In this section, we simulate a control algorithm on a quantum computer in the presence of noisy inputs and use

the results to investigate ideas for attempting to thwart some types of attacks on the control system. We begin with a description of the basic operating principles of a quantum computer. Rather than storing information using bits (represented by 0's or 1's) as classical computers do, quantum computers use quantum bits, or "qubits", to encode information as $|0\rangle$, $|1\rangle$, or in a superposition of the two states $(c_1|0\rangle + c_2|1\rangle)$. Quantum algorithms are implemented through a series of gates, known as quantum gates, which are used to manipulate qubits between different quantum states. Quantum gates can be represented using matrices if qubits are represented via vectors ($|0\rangle = [1\ 0]^T$, $|1\rangle = [0\ 1]^T$); for example, the NOT gate can be represented by the matrix $[0\ 1; 1\ 0]$, because multiplying this matrix by $|0\rangle$ gives back $|1\rangle$, and vice versa. An important unitary operation for placing qubits in a superposition of states is denoted by the "Hadamard matrix", H, and is represented by $H = \frac{1}{\sqrt{2}}[1\ 1; 1\ -1]$ (Yanofsky and Mannucci (2008)). Controlled rotation gates (Z_k gates, where $Z_k = [1\ 0;\ 0\ e^{\frac{2\pi i}{2^k}}]$) can be used as part of an addition strategy with H gates (where the addition strategy is based on a Quantum Fourier Transform (QFT) (Yanofsky and Mannucci (2008); Ruiz-Perez and Garcia-Escartin (2017))).

QFT adds two numbers deterministically, and therefore on a quantum computer with no noise, would produce the same sum of two numbers that a classical computer would. However, if QFT is used in computing a control action (for example, in computing $2x$ as part of the control strategy $u = -2x$, via the sum of x and x), then decoherence and noise in today's quantum computers can decrease the fidelity of the gates and result in different inputs being applied to the process than if there was no noise.

In this section, we explore what happens when the input $u = -2x$, to be applied to the process $\dot{x} = x+u$ (this input is thus stabilizing if implemented deterministically with infinite precision) is computed partially via a quantum simulator (specifically, $x+x$ is computed via a quantum simulator). The simulator (called qasm_simulator) is available from IBM's Quantum Experience with the software development kit Qiskit. A Quantum Fourier Transform-based addition, based on a modified version of Anagolum (2018), is incorporated into a closed-loop simulation for evaluating the input. Noise is accounted for in the algorithm run on the quantum simulator using a custom noise model (depolarizing error) applied with respect to a "cp" gate in the QFT-based addition algorithm. As a result, multiple runs of the addition algorithm (or "shots" as they are called in the IBM Quantum Experience) are run at each sampling time. The case with a single "shot" (or number of iterations that a quantum algorithm is run) used in a computation of u, is compared against the case where the quantum algorithm is run with multiple shots, in addition to the result when evaluated using a classical digital computer. The depolarizing error probability was set to 0.05.

The control input evaluated in the quantum computing framework using QFT-based addition first requires the state measurements of the process sent to the controller, in this case the quantum simulator, to be represented as a combination of $|0\rangle$ and/or $|1\rangle$. This is achieved by first truncating the state measurement to two decimal places at

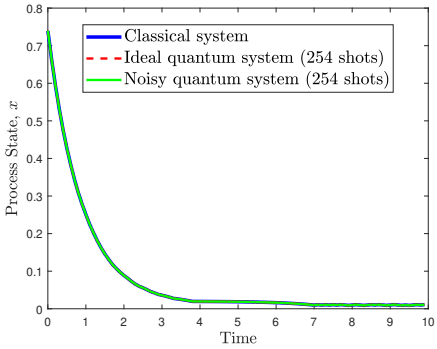

Fig. 2. State trajectories with 254 shots.

every sampling time via multiplication by 100 and taking the floor and absolute value of the consequent value. This number is then converted to its binary representation and placed on the qubits (a NOT gate can be used to flip one of the bits initially in a $|0\rangle$ state to a $|1\rangle$ state). Once the modified state measurement is operated on by QFT-based addition, it is then transformed back to an integer value from binary and divided by 100 to provide the component of the control action, $2x = x + x$. The negative sign is accounted for by negating the result if the value of x was positive. The QFT-based addition algorithm was executed in a manner that requested the size of x in binary before the addition was begun. It then added a 0 in the most significant bit and performed the addition using registers of the resulting size. Ten hours of operation of the process were simulated, with the initial condition $x(0) = 0.74$, using the Explicit Euler numerical integration method with an integration step of 0.001 time units and a sampling period of 0.1 time units. Simulations were performed using Python and the Integrated Development Environment (IDE), Spyder, using both 1 shot and 254 shots for comparison. When the process input is evaluated on a quantum simulator with no noise, the process states and inputs match those from a classical implementation.

The state trajectories for three simulations are compared in Fig. 2, with the quantum simulations being run with 254 shots, and in Fig. 3 with the quantum simulations being run with 1 shot. The classical computer is referred to as the "Classical system," the quantum simulator with the designated number of shots and no noise is the "Ideal quantum system," while with noise it is the "Noisy quantum system." Despite the noise, in both cases, the control actions computed appear stabilizing. Since the impact of the noise is to cause unintended control actions to be applied to the process, this raises the question of what the noise in the quantum computer could mean for cyberattacks.

To investigate this, we first consider the intuitive attack where a false state measurement of -0.02 is continuously provided to the controller when the closed-loop state is initialized at 0.74. If this input is applied, then $u = -2x$ is positive, and $\dot{x} = x+u$ then has a positive right-hand side that will cause x to increase and to drive the closed-loop state away from the origin. To simulate the noisy system in this case, we used a depolarizing error parameter of 1 and a single shot to perturb the control actions compared to the classical computation case. The resulting closed-

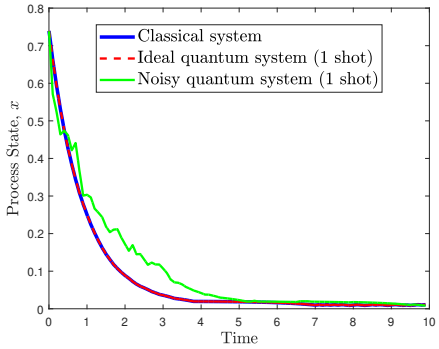

Fig. 3. State trajectories with 1 shot.

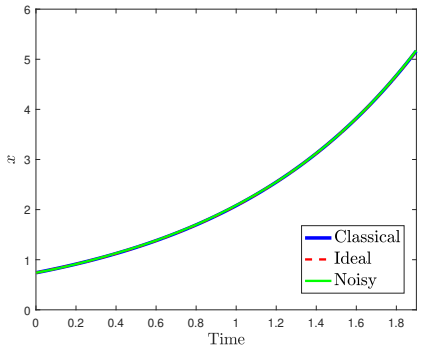

Fig. 4. State trajectories with a false sensor measurement of -0.02 and 1 shot.

loop state and input trajectories are presented in Figs. 4-5 (the case with inputs computed on a classical computer is "Classical," on a quantum simulator with no noise and 1 shot is "Ideal," and on a quantum simulator with 1 shot and a depolarizing error parameter of 1 is "Noisy"). The results with the classical and ideal quantum simulator are overlaid, but the inputs computed by the noisy computer are significantly different. Despite this, we can see that the closed-loop state trajectories are similar. This is due to two major factors: 1) The sign is implemented outside of the addition algorithm, so there is no randomness to the sign that could cause x to decrease under any of the inputs computed by the noisy computer; instead, x will always increase because $\dot{x}$ will always have a positive sign. 2) The size of the qubit registers depends on the state measurement, so the inputs computed by the noisy computer cannot be overly different from those computed by the classical computer when the state measurement is a small value. In this way, despite the noise, the attacker is able to gain approximately the same problematic state trajectory even with differences in the input trajectory compared to that in the classical case.

The analysis above indicated that the noise in the quantum hardware was not beneficial for handling an attack when the sign of the input was fixed in the wrong direction using the state measurement. We could then instead explore what happens if the sign of the input will turn out correctly (i.e., re-running the simulation with the closed-loop state measurement starting at 0.74 and the false state measurement at 0.02). However, in this case, a similar

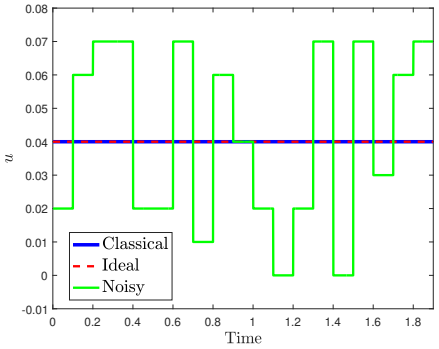

Fig. 5. Input trajectories with a false sensor measurement of -0.02 and 1 shot.

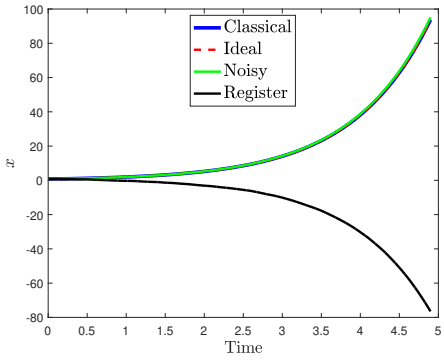

Fig. 6. Comparison between the state trajectories with the false sensor measurement of 0.02. "Ideal" and "Noisy" use a shrinking register; "Register" uses a fixed register size with 1 shot and a depolarizing error parameter 1.

phenomenon is observed where because the register size in the addition procedure is limited, even in the noisy simulation, no input is computed that has a large enough magnitude for its absolute value to cause $\dot{x}$ to be negative when the initial value of the state is 0.74.

We could ask then if using a fixed register size could help to alleviate this problem or not, as then in the case that inputs are randomly selected, there are more possibilities. In this case, we use a register size fixed to the size that would be obtained if the state measurement was 0.74. The result for the closed-loop state is shown in Fig. 6. While the use of the fixed register size creates different inputs than in the case with the shrinking register size, it runs into the issue that when the actual closed-loop state becomes negative, since the input being computed for the false state measurement of 0.02 is negative, it is able to continue to cause the closed-loop state to decrease. Even if the depolarizing error parameter is decreased to 0.05, the sign is still not able to flip since the false state measurement is always positive. Again the closed-loop state continues to decrease after x becomes negative under the sample-and-hold control law.

The results above help to clarify that despite that the state trajectories in Fig. 3 appear stabilizing in the presence of the noise, this is not an indication that an arbitrarily bad state measurement could be obtained and that the

inputs would still be stabilizing; for example, the sign of the state measurement was still correct. We also see that modification to the register size alone is unlikely to be able to remove all potentially bad inputs for a given state measurement from the set of possible inputs. Though the noise in the quantum computer was not able to fight the cyberattacks on its own, the discussion of the implementation of the addition algorithm does provide a potential concept for probing for attacks via the detailed implementation of a control law. For example, consider an attack where the actual state measurement is 0.74, but the false state measurement is 0.36 (so that $-2x$ is -0.72, which will cause $\dot{x}$ for the actual system to be positive). In the absence of noise, the control action computed would appear to be stabilizing, and an attacker could provide a state trajectory that appears consistent with the control law believed to be computed (which would be driving the closed-loop state toward the origin). However, suppose that at the time when the falsified state measurement reads 0.36, the size of the register is capped and a control action that would not be stabilizing if the actual state was 0.36 is applied. If the state measurement continues to decrease x, this would indicate an attack.

Remark 1. This section does not focus on computational efficiency of control implemented on quantum computers; in general, developing quantum computing algorithms with benefits compared to classical computing algorithms can be challenging, so that past works (e.g., Cincio et al. (2018)) have even suggested using machine learning to develop quantum algorithms. However, there are many considerations to be taken into account when attempting to do something like this. To see this, we can consider a thought experiment in using optimization to seek to computationally design a means to use quantum computing to help with locating the ground state energy of a material, for which algorithms such as the variational quantum eigensolver (VQE) (Moll et al. (2018)) have been used. We can consider a case where an attempt is made to locate a quantum algorithm that, in combination with post-processing of data from the qubits using a classical computer, uses a single qubit and four gates (in the absence of noise). If we consider IBM's ibmq_manila, the available gates are CX, ID, RZ, SX and X. Of these, the single-qubit gates are ID, SX, X, and RZ, corresponding to $ID = [1\ 0; 0\ 1]$, $SX = \frac{1}{2}[1+i\ 1-i; 1-i\ 1+i]$, $X = [0\ 1; 1\ 0]$ and $RZ = [e^{\frac{-i\lambda}{2}}\ 0; 0\ e^{\frac{i\lambda}{2}}]$ where λ is a parameter. These gates will take a qubit initially in state $|0\rangle$ to the final state $\alpha|0\rangle + \beta|1\rangle$ where the post-processing algorithm will set the energy $E = c_1\alpha + c_2\beta$. c_1 and c_2 will be continuous decision variables to be determined by the optimization problem that is "searching" for an algorithm of the class specified to attempt to match the algorithm result to data on ground state energies. Discrete decision variables (related to the selection and position of the mentioned gates to be applied in the circuit) should also be included. The objective function of the optimization problem will be considered to be: $\sum_{k=1}^{N}(E_{pred,k} - E_{actual,k})^2$, where N is the number of molecules in the set being used to find (train) the algorithm, and $E_{pred,k}$ is the value of E for the k-th molecule for a given value of the decision variables. $E_{actual,k}$ are considered to be the values obtained from a data set. Because the actions of gates on qubits can be described by matrices, the final values of α_k and β_k (i.e., α and β for the k-th molecule) can be obtained by multiplying $|0\rangle = [1\ 0]^T$ by a series of four matrices representing the gates. We consider an *ad hoc* selection of chemistry properties for dictating some of the gates to be used (to make the algorithm chemistry-dependent). For example, we can consider the first two gates which impact the qubits to represent whether certain chemistry properties (e.g., types of bonds) are present (properties $p1$ and $p2$). The decision variables indicating which gates are selected on the first two gates impacting the qubits are represented by δ_{3ikp1} and δ_{4ikp2}, $i = 1, 2, 3, 4$. The last two gates to act on the qubit are taken to be the same for all of the molecules (i.e., they are not chemistry-dependent) and are selected by binary variables $\delta_{ijk} \in [0, 1]$, where $i \in [1, 2]$ corresponds to the position of the gate (with 1 signifying the last gate and 2 the second-to-last gate), k signifies the k-th molecule, and $j \in [1, 2, 3, 4]$ signifies the gate's type (ID, RZ, X, or SX). This would lead to the following expressions for α_k and β_k, along with requirements that only one gate can be selected for each position:

$$\begin{aligned}\begin{bmatrix}\alpha_k\\ \beta_k\end{bmatrix} = [[\delta_{11k}ID + \delta_{12k}RZ + \delta_{13k}X + \delta_{14k}SX]*\\ [\delta_{21k}ID + \delta_{22k}RZ + \delta_{23k}X + \delta_{24k}SX]*\\ [\delta_{31kp1}ID + \delta_{32kp1}RZ + \delta_{33kp1}X + \delta_{34kp1}SX]*\\ [\delta_{41kp2}ID + \delta_{42kp2}RZ + \delta_{43kp2}X + \delta_{44kp2}SX]] * \begin{bmatrix}1\\ 0\end{bmatrix}\end{aligned} \tag{5}$$

$$\delta_{11k} + \delta_{12k} + \delta_{13k} + \delta_{14k} = 1 \tag{6}$$

$$\delta_{21k} + \delta_{22k} + \delta_{23k} + \delta_{24k} = 1 \tag{7}$$

$$\delta_{31kp1} + \delta_{32kp1} + \delta_{33kp1} + \delta_{34kp1} = 1 \tag{8}$$

$$\delta_{41kp2} + \delta_{42kp2} + \delta_{43kp2} + \delta_{44kp2} = 1 \tag{9}$$

This strategy provides a means to set up a learning strategy, but is unlikely to provide a useful circuit. If the data includes two molecules with the same chemistry properties $p1$ and $p2$, α_k and β_k will be the same. This restricts the possible values that can be returned as $c_1\alpha_k + c_2\beta_k$. This reflects some of the challenges of coming up with new algorithms on a quantum computer that are beneficial using computational techniques; the learning algorithms need to have a search space that includes non-classical manipulations that could be significant.

4 Conclusion and Perspectives on Design for Resilience Against Cyberattacks

In this work, motivated by simulations of a simple control law implemented on a quantum simulator with noise where the applied control actions appeared to be stabilizing despite uncertainty, we investigated what these results for quantum computing-implemented control indicate for control system cybersecurity (there was not a clear benefit to quantum computers' randomness for attempting to thwart cyberattacks in the simulations studied). While modifying control actions to attempt to handle attacks, as was done in the course of the quantum computing-implemented control study, is one way of attempting to address cyberattacks on control systems, process design is another way, and may be particularly meaningful in cases where failure to detect an attack could otherwise be catastrophic. For example, consider a geothermal plant that is a renewable source of heat. Prior work, such as Mohan et al. (2015) analyzed an enhanced geothermal system coupled

with the binary Organic Rankine Cycle (ORC) for power extraction with a variety of working fluids. This was used to compare success metrics such as power output. However, different design decisions may have different impacts not only on the power output, but also on how much it changes with changes in other process variables. Considering the robustness of the power output to changes that an attacker might make to the system could be a consideration in the design of a system particularly for cases such as more remote locations (e.g., island nations such as Comoros where failure of an energy system could make obtaining back-up energy solutions more challenging due to the lack of connection to other land resources). In addition to considering resilience in process design, resilience can also be considered in design of systems in other domains where process systems engineers might apply their expertise. For example, consider a case where optimization is used for organizing volunteers for nonprofits. Since volunteers fulfill an essential role in communities around the world by offering support to nonprofit organizations, social causes could be disrupted as a consequence of misappropriation of resources by an optimizer. As a thought experiment, consider the problem of helping volunteers to find the right nonprofit support in a given week to make the most impact. An optimization problem might be formulated in which a number of volunteers is split between several nonprofits, with constraints based on the number of hours they are available. One could imagine a problem for this optimal resource allocation problem if "fake" volunteers state that they will put in hours to skew the scheduling, and have no intent of showing up. Strategies should be considered for attempting to thwart malicious behavior to this effect; for example, volunteers may need to be validated or otherwise may be randomly assigned, instead of being considered as contributors with respect to the optimal resource allocation, until they have a track record of showing up for their commitments.

Acknowledgements

Financial support from the Air Force Office of Scientific Research (award number FA9550-19-1-0059), National Science Foundation CNS-1932026 and CBET-1839675, Wayne State University Grants Boost funding, and Wayne State University is gratefully acknowledged. We wish to thank Paul M. Alsing of the Air Force Research Laboratory, Information Directorate (AFRL/RI) for useful comments and discussions. The authors would like to acknowledge sources that were helpful in preparing this paper: 1) the YouTube videos on QFT and QFT-based addition (`https://quantumguru.net/quantumalgorithms/quantumalgorithms.html`); 2) "A practical introduction to quantum computing" by Elias Fernandez-Combarro Alvarez and CERN. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. We thank Naseem Abou-Ghaida for analyzing a geothermal energy optimization reference.

Literature Cited

Ajagekar, A., Humble, T., and You, F. (2020). Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. *Computers & Chemical Engineering*, 132, 106630.

Ajagekar, A. and You, F. (2019). Quantum computing for energy systems optimization: Challenges and opportunities. *Energy*, 179, 76–89.

Alanqar, A., Ellis, M., and Christofides, P.D. (2015). Economic model predictive control of nonlinear process systems using empirical models. *AIChE Journal*, 61, 816–830.

Alessandri, A., Baglietto, M., Battistelli, G., and Zavala, V. (2010). Advances in moving horizon estimation for nonlinear systems. In *IEEE Conference on Decision and Control*, 5681–5688. Atlanta, Georgia.

Anagolum, S. (2018). Donew. `https://github.com/SashwatAnagolum/DoNew`.

Cincio, L., Subaşı, Y., Sornborger, A.T., and Coles, P.J. (2018). Learning the quantum algorithm for state overlap. *New Journal of Physics*, 20, 113022.

Durand, H. (2018). A nonlinear systems framework for cyberattack prevention for chemical process control systems. *Mathematics*, 6(9), 169.

Mohan, A.R., Turaga, U., Subbaraman, V., Shembekar, V., Elsworth, D., and Pisupati, S.V. (2015). Modeling the CO2-based enhanced geothermal system (EGS) paired with integrated gasification combined cycle (IGCC) for symbiotic integration of carbon dioxide sequestration with geothermal heat utilization. *International Journal of Greenhouse Gas Control*, 32, 197–212.

Moll, N., Barkoutsos, P., Bishop, L.S., Chow, J.M., Cross, A., Egger, D.J., Filipp, S., Fuhrer, A., Gambetta, J.M., Ganzhorn, M., Kandala, A., Mezzacapo, A., Müller, P., Riess, W., Salis, G., Smolin, J., Tavernelli, I., and Temme, K. (2018). Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3, 030503.

Oyama, H., Messina, D., Rangan, K.K., and Durand, H. (2022). Lyapunov-based economic model predictive control for detecting and handling actuator and simultaneous sensor/actuator cyberattacks on process control systems. *Frontiers in Chemical Engineering*.

Oyama, H. and Durand, H. (2020). Integrated cyberattack detection and resilient control strategies using lyapunov-based economic model predictive control. *AIChE Journal*, 66(12), e17084.

Ruiz-Perez, L. and Garcia-Escartin, J.C. (2017). Quantum arithmetic with the quantum fourier transform. *Quantum Information Processing*, 16(6), 152.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25–57.

Walther, A. (2010). `https://projects.coin-or.org/ADOL-C/browser/trunk/ADOL-C/examples/additional_examples/ipopt/MittelmannDistCntrlNeumA?rev=78`.

Walther, A. and Griewank, A. (2009). Getting started with ADOL-C. *Combinatorial Scientific Computing*, (09061), 181–202.

Yanofsky, N.S. and Mannucci, M.A. (2008). *Quantum Computing for Computer Scientists*. Cambridge University Press.