Adiabatic Quantum Linear Optimal Control for Discrete Time Dynamical Systems

Mayank-Shekhar Jha
CNRS, CRAN
Universite de Lorraine
Nancy F-54000, France
mayank-shekhar.jha@univ-lorraine.fr

Prasanna Date
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge 37830, USA
datepa@ornl.gov

***Abstract*—This paper presents a novel approach for solving finite-horizon linear optimal control based on block-Toeplitz least-squares problem, by recasting the latter as Quadratic Unconstrained Binary Optimization (QUBO) suitable for adiabatic quantum computing (AQC). Classical block-Toeplitz least-squares problem, scales as $O((Nm)^3)$ for horizon length N and control dimension m. We demonstrate that this Toeplitz-structured least-squares cost can be transformed into a QUBO formulation by introducing binary precision vectors that encode each continuous control parameter into finite number of bits. We establish the general case in mathematically rigorous manner wherein the total number of binary decision variables remain dependent on N. To respect current quantum annealer capabilities, we propose a basis-function parametrization that approximates the full control sequence with a small set of basis coefficients, reducing the total number of binary-variables and rendering the latter independent of N. Through simulation study, we show that, for large datasets, the QUBO pipeline, comprising of hardware-constant anneal step outperforms classical least square based solvers by several factors whilst demonstrating acceptable accuracy.**

***Index Terms*—adiabatic quantum computing, optimal control, quantum control, least squares**

I. Introduction

Optimal control theory is a fundamental aspect of automatic control, focusing on designing control strategies that guide dynamic systems toward desired objectives by optimizing specified performance criterion(s), indispensable across various engineering fields [8, 10] . Such formulations enable leveraging well-established mathematical optimization methods, including linear algebra and least squares optimization [2]. For linear discrete-time systems, the control task typically involves determining an action sequence that minimizes a quadratic cost reflecting penalties on state deviations and control inputs [10]. Traditionally, this problem is formulated and solved as a least squares optimization problem, leveraging linear algebraic methods such as matrix pseudo-inverses and direct solvers [2]. However, classical computation of these solutions, particularly for large-scale or infinite-horizon problems, remains computationally expensive, often suffering from high computational complexity, specifically scaling cubically with the size of data involved, thus becoming prohibitive for large-scale or real-time control tasks [7]. This scaling issue limits their applicability in contemporary scenarios involving highly sampled dynamical systems, frequently encountered in real-time applications that generate substantial amounts of data, demanding rapid and computationally efficient algorithms. Classical computational methods often scale cubically with data size[10], rendering them impractical in many contemporary engineering scenarios that require swift computation and real-time responsiveness.

In recent years, quantum computing, particularly adiabatic quantum computing (AQC)[1], has emerged as a promising paradim for tackling complex optimization problems [6]. AQC exploits quantum mechanical phenomena to approximate solutions to optimization tasks, promising enhanced computational efficiency over classical counterparts. Central to the implementation of AQC is the Quadratic Unconstrained Binary Optimization (QUBO) formulation, known for encapsulating a broad class of combinatorial and continuous optimization problems effectively [9]. Using AQC via QUBO formulations, have demonstrated considerable promise in accelerating machine learning tasks such as linear regression [5, 11]. Recent empirical studies validate significant computational speedups for linear regression tasks when reformulated as QUBO problems solved on quantum annealers like the D-Wave quantum computer [4, 5]. This approach involves encoding continuous optimization problems into binary variables through a precision vector mechanism, significantly enhancing scalability and efficiency [5].

However, optimal control problems, have yet to be explored within this quantum computing framework. Addressing this scientific gap, the current paper proposes, for the first time, a reformulation of finite-horizon linear optimal control problem as QUBO problems solvable via AQC. By capitalizing on the efficiency and scalability of QUBO formulations within quantum computational architectures, this work aims to provide a novel and computationally advantageous method for solving control problems traditionally limited by classical computational constraints.

Thus, main contribution of this paper lies in proposition of a novel QUBO-based formulation suitable for AQC, for linear finite-horizon optimal control problem that is traditionally based on least square approach. In particular, we consider the Toeplitz-matrix-based least square approach and map the quadratic cost function and system dynamics into a binary optimization framework, thus enabling the use of AQC to solve these problems. We establish the novel formulation in a mathematically rigorous manner. We evaluate the proposed

979-8-3315-5736-2/25/$31.00 ©2025 IEEE
DOI 10.1109/QCE65121.2025.10341

method's theoretical advantages in terms of computational complexity, emphasizing its scalability and potential speed-ups over classical least squares-based methods. The findings presented herein serve not only as a pioneering step in integrating AQC approaches into control theory but also as a practical guide for future implementations of quantum-enhanced optimal control algorithms.

Following this section, Section 2 covers background on Toeplitz-based optimal control, QUBO formulation, and problem statement. Section 3 presents the QUBO formulation for finite-horizon control followed by Section 4 that describes basis-function parametrization and provides mathematically rigorous theorems. Section 5 analyzes computational complexity and Section 6 shows shows simulation study followed by Section 7 that concludes and outlines future work.

II. Background and Problem Statement

We consider a linear discrete-time linear system in time invariant state-space form as:

$$x_{k+1} = A x_k + B u_k \tag{1}$$

where $k \in \mathbb{Z}_+$ is the discrete time step, $x_k \in \mathbb{R}^n$ is the state vector at time k, $u_k \in \mathbb{R}^m$ is the control input. $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are known constant matrices. It is assumed that (A,B) is stabilizable, i.e., there exists a feedback control law or policy such that system in closed loop is stable. Moreover, the system is considered fully observable.

A. Finite horizon Optimal Control as a Least Square Problem

In the classic finite horizon optimal control setting, the goal is to find a sequence of control inputs $\{u_0,\dots,u_{N-1}\}$ that minimizes the finite-horizon quadratic cost:

$$J(x_0,\{u_k\}) \;=\; x_N^T Q_f x_N \;+\; \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k\right), \tag{2}$$

where $Q, Q_f \succeq 0$ and $R \succ 0$. The horizon length is N. The final state cost $x_N^T Q_f x_N$ captures terminal penalties, and Q,R weight state and input penalties at each stage. Consider the system variables along a trajectory consisting of N time steps and define the stacked vectors as $U = \begin{bmatrix} u_0 & u_1 & \cdots & u_{N-1}\end{bmatrix}^\top \in \mathbb{R}^{Nm}$ and $X = \begin{bmatrix} x_0 & x_1 & \cdots & x_N\end{bmatrix}^\top \in \mathbb{R}^{(N+1)n}$. Then, from (1), it follows that:

$$X = H x_0 + G U, \tag{3}$$

where $H \in \mathbb{R}^{(N+1)n\times n}$ models the evolution due to the initial condition, and $G \in \mathbb{R}^{(N+1)n\times Nm}$ is a *block-Toeplitz* matrix captures how control inputs affect all future states, with

$$H = \begin{bmatrix} I & A & A^2 & \cdots & A^N \end{bmatrix}^\top, \tag{4}$$

$$G = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}. \tag{5}$$

In other words, each block-subdiagonal of G is $A^{i-1}B$, making G block-Toeplitz. Rewriting the cost (2) as $J(U)$ in a block-matrix form gives:

$$\begin{aligned} J(U) \;=\; &\|\mathrm{diag}(Q^{1/2},\dots,Q^{1/2},Q_f^{1/2})\cdot X\|_2^2 \\ &+ \|\mathrm{diag}(R^{1/2},\dots,R^{1/2})\cdot U\|_2^2, \end{aligned} \tag{6}$$

with $X = Hx_0 + GU$. Let

$$Q_{\text{block}} = \mathrm{blockdiag}(Q,\dots,Q,Q_f), \quad R_{\text{block}} = \mathrm{blockdiag}(R,\dots,R), \tag{7}$$

Then, substituting X from (3) into (6) leads to the typical least squares form:

$$J(U) = U^T \underbrace{(G^T Q_{\text{block}}\, G + R_{\text{block}})}_{=:M} U \,+\, 2U^T \underbrace{(G^T Q_{\text{block}}\, H x_0)}_{=:f} + \delta. \tag{8}$$

where δ is a real constant. The optimal control inputs U^* can be obtained by solving:

$$U^* = \arg\min_{u\in U} \left(U^T M U + 2U^T f\right). \tag{9}$$

Computing first gradient of $J(U)$ with respect to U and equating it to zero, reduces the optimal control to solving the following equation:

$$M U^* = -f, \quad \text{where } M \in \mathbb{R}^{Nm\times Nm}. \tag{10}$$

Because G is block-Toeplitz, this "Toeplitz-matrix-based" formulation reduces the linear-quadratic optimal control on a finite horizon, fundamentally, to a *least squares* problem in Nm dimensions. Solving $MU = -f$ by dense inversion typically costs $O((Nm)^3)$.

B. Quadratic Unconstrained Binary Optimization (QUBO) Problem

A *Quadratic Unconstrained Binary Optimization* (QUBO) problem is defined as:

$$\min_{\hat{z}\in\{0,1\}^{N_{qb}}} \; \hat{z}^T \mathscr{A}\, \hat{z} \,+\, \mathscr{B}^T \hat{z}, \tag{11}$$

where $\mathscr{A} \in \mathbb{R}^{N_{qb}\times N_{qb}}$ is a symmetric matrix and $\mathscr{B} \in \mathbb{R}^{N_{qb}}$ where N_{qb} is the number of binary decision variables. Each decision variable is binary $\hat{z}_i \in \{0,1\}$. An optimization problem with decision variables restricted to $\{0,1\}$ and objective function quadratic can be cast in this form. A critical step is mapping each real variable (for instance, control inputs) to a finite set of binary variables. To that end, a *precision vector* $\mathscr{P} \in \mathbb{R}^L$ [5] with $p_i, i = 1,2,...L$ as real elements, is considered as:

$$\mathscr{P} \;=\; [p_1,\, p_2,\, \dots,\, p_L]^T. \tag{12}$$

$\mathscr{P}$ is generally chosen to reflect a desired numeric range and resolution. For example: $\mathscr{P} = [-1,\, -\frac{1}{2},\, \frac{1}{2},\, 1]$ can be used to represent signed increments from -1.5 to $+1.5$ in increments of 0.5. In general, a precision vector $\mathscr{P}$ allows for encoding of a real continuous variable(s) w_i using binary variables $\hat{w}_{ik} \in \{0,1\}$ as

$$w_i \approx \sum_{t=1}^{L} p_t \, \hat{w}_{it}, \quad \hat{w}_{it} \in \{0,1\}. \tag{13}$$

where $\{p_1, \ldots, p_L\}$ are chosen to achieve the desired precision (e.g., $p_t = 2^{t-1}$ or any scaled version). It is noted that for Nm real control variables, the total number of binary variables becomes $N_{qb} = N\ m\ L$.

C. Adiabatic Quantum Computing

In AQC, one encodes the QUBO cost (11) as a Hamiltonian:

$$\mathscr{H}(\hat{z}) = \hat{z}^T \mathscr{A} \, \hat{z} + \mathscr{B}^T \hat{z}. \tag{14}$$

The quantum system is initiated in the *ground state* of a simpler Hamiltonian [9] and then evolved slowly towards (14)[6]. Under the *adiabatic theorem* [3], if the evolution is sufficiently slow, the system remains in the lowest-energy configuration, which ideally corresponds to the global optimum of the QUBO [1].

The following sections present the contributions of the paper.

III. QUBO based finite horizon optimal control

Consider the finite horizon optimal control problem as presented in (2)-(10). In this section, we show how such a problem can be transformed to a QUBO one, suitable for AQC. From (8), let $M = G^T Q_{\text{block}} G + R_{\text{block}}$ and $f = G^T Q_{\text{block}} H x_0$. Then,

$$J(U) = U^T M U + 2 f^T U + \delta. \tag{15}$$

To encode the ith control input u_i, we introduce corresponding binary variables $\{\hat{u}_{i1}, \hat{u}_{i2}, \ldots, \hat{u}_{iL}\}$ each in $\{0,1\}$ where $\forall t \in [0,L], \hat{u}_{it}$ corresponds to the binary encoding of the i-th entry of U with L precision bits. Then, u_i is approximated by:

$$u_i \approx \sum_{t=1}^{L} p_t \, \hat{u}_{it}, \quad \hat{u}_{it} \in \{0,1\}. \tag{16}$$

Hence, each real variable u_i is replaced by L binary variables. Considering (17), Nm real control input variables lead to N_{qb} binary decision variables where $N_{qb} = L \times Nm$. Now consider a single binary vector $\hat{U} \in \{0,1\}^{N_{qb}}$ that stacks, $\forall u_i \in U \in \mathbb{R}^{Nm}$, all the binary variables $\hat{u}_{it}$ as:

$$\hat{U} = \begin{bmatrix} \hat{u}_{11} & \hat{u}_{12} & \ldots & \hat{u}_{1L} & \hat{u}_{21} & \ldots & \hat{u}_{NmL} \end{bmatrix}^T, \tag{17}$$

Next, consider a block-diagonal matrix $P_{\text{block}} \in \mathbb{R}^{Nm \times N_{qb}}$ such that $P_{\text{block}} = \text{diag}(P, P, \ldots, P) \in \mathbb{R}^{Nm \times N_{qb}}$, constructed by replicating P for each entry of U, we have:

$$U \approx P_{\text{block}} \hat{U} \tag{18}$$

As such, the approximation $U \approx P_{\text{block}} \hat{U}$ allows mapping binary variables $\hat{U}$ to real values U.

Theorem 1 (Finite-Horizon Optimal Control as QUBO) *Let the finite-horizon least-squares optimal control problem be obtained by minimizing* (8) *where $U \in \mathbb{R}^{Nm}$. Suppose U is encoded into a binary vector $\hat{U} \in \{0,1\}^{N_{qb}}$ with $U = P\hat{U}$ where $P \in \mathbb{R}^{Nm \times N_{qb}}$ is a (block) matrix. If there exists a matrix $\overline{\mathscr{A}} \in \mathbb{R}^{N_{qb} \times N_{qb}}$ and a vector $\overline{\mathscr{B}} \in \mathbb{R}^{N_{qb}}$ then, the classic finite horizon optimal control problem* (9) *can be expressed in canonical QUBO form as:*

$$\min_{\hat{U} \in \{0,1\}^{N_{qb}}} \hat{U}^T \overline{\mathscr{A}} \, \hat{U} + \overline{\mathscr{B}}^T \hat{U}, \tag{19}$$

From (8) we have $J(U) = U^T M U + 2 f^T U + \delta$ where $M \in \mathbb{R}^{Nm \times Nm}$ and $f \in \mathbb{R}^{Nm}$. Using the binary encoding as shown in (17), we have $U = P\hat{U}, \hat{U} \in \{0,1\}^{N_{qb}}$. Substituting U in the expression for J yields,

$$U^T M U = (\hat{U}^T P^T) M (P\hat{U}) = \hat{U}^T (P^T M P) \hat{U}. \tag{20}$$

Similarly, for the linear term,

$$2 f^T U = 2 f^T (P\hat{U}) = (2 P^T f)^T \hat{U}. \tag{21}$$

Dropping constant terms that do not affect the minimizer, the cost can be written as

$$\hat{U}^T (P^T M P) \hat{U} + (2 P^T f)^T \hat{U}. \tag{22}$$

Hence, defining $\overline{\mathscr{A}} = P^T M P$ and $\overline{\mathscr{B}} = 2 P^T f$ transforms the minimization problem of $J(U)$ over continuous U into a canonical QUBO form (11) as:

$$\min_{\hat{U} \in \{0,1\}^{N_{qb}}} \hat{U}^T \overline{A} \hat{U} + \overline{\mathscr{B}}^T \hat{U}. \tag{23}$$

Consequently, the original finite-horizon least-squares optimal control problem is reformulated as a QUBO problem.

IV. Basis-Function Parameterization for QUBO-Based Control

In the fully parameterized QUBO approach of Section III, each scalar control $u_{k,i}$ at time k (for $i = 1, \ldots, m$, $k = 0, \ldots, N-1$) was encoded by L binary bits. As a result, the total binary-variable count become $N_{qb} = N \times m \times L$, dependent on the number of data N. The typical D-Wave based hardware limit remains at 64 binary variables [5] i.e. $N \times m \times L < 64$,. The latter poses a great challenge to the proposed formulation restricting maximum value of N to 64.

To mitigate this, we now propose a reduced-dimensional parameterization that uses r basis functions (with $r \ll N$). By expressing the entire control sequence through these r basis vectors, we reduce and binary-encode only rm continuous variables, thereby decoupling dependence of total binary variables on N. The approach is formalized as follows.

A. Basic Setup

Consider each $u_{k,i}$ encoded by L bits, that would require $N_{qb} = NmL$ binary variables. To mitigate dependency on N, we choose a set of r scalar basis functions $\{\phi_j(k)\}_{j=1}^r$, $k = 0, \ldots, N-1$, and propose:

$$u_k = \sum_{j=1}^{r} \alpha_j \, \phi_j(k), \quad k = 0, 1, \ldots, N-1, \tag{24}$$

where each $\alpha_j \in \mathbb{R}^m$ is an m-vector of basis coefficients (independent of k). We then *stack* those coefficient-vectors into a single vector $\Theta = \begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_r\end{bmatrix}^\top \in \mathbb{R}^{rm}$. To relate Θ back to the full control stack $U \in \mathbb{R}^{Nm}$, define the time-basis matrix

$$\Phi = \begin{bmatrix} \phi_1(0) & \phi_2(0) & \cdots & \phi_r(0) \\ \phi_1(1) & \phi_2(1) & \cdots & \phi_r(1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(N-1) & \phi_2(N-1) & \cdots & \phi_r(N-1) \end{bmatrix} \in \mathbb{R}^{N\times r},$$

and set $\Psi = \Phi \otimes I_m \in \mathbb{R}^{(Nm)\times(rm)}$. Then by stacking all u_k, one gets:

$$U = \begin{bmatrix}u_0 & u_1 & \cdots & u_{N-1}\end{bmatrix}^\top = \Psi\Theta. \tag{25}$$

with $U \in \mathbb{R}^{Nm}$, $\Theta \in \mathbb{R}^{rm}$. Hence, instead of minimizing $J(U)$ over $U \in \mathbb{R}^{Nm}$, one minimizes

$$J(\Psi\Theta) = (\Psi\Theta)^T M (\Psi\Theta) + 2 f^T (\Psi\Theta) + \text{const.}$$

over $\Theta \in \mathbb{R}^{rm}$. Substituting $U = \Psi\Theta$ into the original cost and defining $\widetilde{M} = \Psi^T M \Psi \in \mathbb{R}^{rm\times rm}$ and $\widetilde{f} = \Psi^T f \in \mathbb{R}^{rm}$ leads to

$$\begin{aligned} J(\Psi\Theta) &= \Theta^\top(\Psi^\top M\Psi)\Theta + 2(\Psi^\top f)^\top\Theta + \text{constant} \\ &= \Theta^\top \widetilde{M}\Theta + 2\widetilde{f}^\top\Theta + \text{const.} \end{aligned} \tag{26}$$

reducing the problem to

$$\min_{\Theta\in\mathbb{R}^{rm}} J_{\text{red}}(\Theta) = \Theta^\top \widetilde{M}\Theta + 2\widetilde{f}^\top\Theta + c_1 \tag{27}$$

where c_1 is any constant. Setting $\nabla_\Theta J_{\text{red}} = 0$ yields

$$\widetilde{M}\Theta^* = -\widetilde{f}, \quad \widetilde{M} \in \mathbb{R}^{rm\times rm}, \ \widetilde{f} \in \mathbb{R}^{rm}.$$

Now, Θ^* can be obtained by solving the standard rm-dimensional linear-solve/least-squares problem.

B. Binary Encoding of the Coefficient Vector

To transform this into a QUBO, we now encode each of the rm real entries of Θ using L bits. Fix a precision-vector $\mathscr{P} = \begin{bmatrix}p_1 & p_2 & \cdots & p_L\end{bmatrix}^\top \in \mathbb{R}^L$, so that each scalar component $(\alpha_j)_i$ (the i-th entry of $\alpha_j \in \mathbb{R}^m$) is approximated by

$$(\alpha_j)_i \approx \sum_{t=1}^{L} p_t \widehat{\alpha}_{ijt}, \qquad \widehat{\alpha}_{ijt} \in \{0,1\}. \tag{28}$$

Since Θ stacks $\{\alpha_1,\ldots,\alpha_r\}$, there are rm scalar entries to encode, each requiring L bits. We then stack all these bits into a single binary vector as $\widehat{\Theta} = \begin{bmatrix}\widehat{\alpha}_{111} & \cdots & \widehat{\alpha}_{11L} & \widehat{\alpha}_{211} & \cdots & \widehat{\alpha}_{21L} & \cdots & \widehat{\alpha}_{mr1} & \cdots & \widehat{\alpha}_{mrL}\end{bmatrix}^\top \in \{0,1\}^{rmL}$.

Let us define the block-diagonal "precision matrix" as

$$P_{\text{param}} = \operatorname{diag}(P, P, \ldots, P) \in \mathbb{R}^{rm\times(rmL)}, \tag{29}$$

where each diagonal block $P \in \mathbb{R}^{m\times(mL)}$ is itself structured so that $\alpha_j = \begin{bmatrix}(\alpha_j)_1 & (\alpha_j)_2 & \cdots & (\alpha_j)_m\end{bmatrix}^\top$. Then, consider

$$\alpha_j \approx P\begin{bmatrix}\widehat{\alpha}_{1j1} & \cdots & \widehat{\alpha}_{1jL}\widehat{\alpha}_{2j1} & \cdots & \widehat{\alpha}_{2jL}\cdots & \widehat{\alpha}_{mjL}\end{bmatrix}^\top, \tag{30}$$

with $P = I_m \otimes \mathscr{P}^\top$. Finally, stacking over $j = 1,\ldots,r$,

$$\Theta \approx P_{\text{param}}\widehat{\Theta}, \qquad \widehat{\Theta} \in \{0,1\}^{rmL}. \tag{31}$$

We now state and prove the main theorem: under the basis-function parametrization (24), minimizing the original finite-horizon cost is equivalent to a QUBO in $\widehat{\Theta} \in \{0,1\}^{rmL}$.

Theorem 2 (Basis-Function Parameterized QUBO problem) *Let*

$$\begin{aligned} J(U) &= U^\top MU + 2f^\top U + \delta, \\ \textit{with} \quad M &= G^\top Q_{\text{block}}G + R_{\text{block}}, \\ f &= G^\top Q_{\text{block}}Hx_0. \end{aligned} \tag{32}$$

Suppose the control sequence is approximated by $U = \Psi\Theta$ with $\Psi = \Phi \otimes I_m$ and $\Theta \in \mathbb{R}^{rm}$. Define $\widetilde{M} = \Psi^T M\Psi \in \mathbb{R}^{rm\times rm}$, $\widetilde{f} = \Psi^T f \in \mathbb{R}^{rm}$. If each entry of Θ is then encoded via $\widehat{\Theta} \in \{0,1\}^{rmL}$ as in (31), *then minimizing $J(U)$ can be formulated as a QUBO problem :*

$$\min_{\widehat{\Theta}\in\{0,1\}^{rmL}} \left[\widehat{\Theta}^T \overline{\mathscr{A}}\,\widehat{\Theta} + \overline{\mathscr{B}}^T\widehat{\Theta}\right], \tag{33}$$

where

$$\begin{aligned} \overline{\mathscr{A}} &= P_{\text{param}}^\top \widetilde{M} P_{\text{param}} = P_{\text{param}}^\top(\Psi^\top M\Psi)P_{\text{param}}, \\ \overline{\mathscr{B}} &= 2P_{\text{param}}^\top \widetilde{f} = 2P_{\text{param}}^\top(\Psi^\top f). \end{aligned} \tag{34}$$

Starting from the reduced cost:

$$J_{\text{red}}(\Theta) = \Theta^T \widetilde{M}\Theta + 2\widetilde{f}^T\Theta + \delta.$$

Substitute $\Theta = P_{\text{param}}\widehat{\Theta}$. Then

$$\Theta^T \widetilde{M}\Theta = (P_{\text{param}}\widehat{\Theta})^T \widetilde{M}(P_{\text{param}}\widehat{\Theta}) = \widehat{\Theta}^T(P_{\text{param}}^T\widetilde{M}P_{\text{param}})\widehat{\Theta},$$

and

$$2\widetilde{f}^T\Theta = 2\widetilde{f}^T(P_{\text{param}}\widehat{\Theta}) = (2P_{\text{param}}^T\widetilde{f})^T\widehat{\Theta}.$$

Dropping the constant δ does not affect the minimizer. It follows that

$$J(\widehat{\Theta}) = \widehat{\Theta}^T(P_{\text{param}}^T\widetilde{M}P_{\text{param}})\widehat{\Theta} + (2P_{\text{param}}^T\widetilde{f})^T\widehat{\Theta} + \text{constant}.$$

Defining $\overline{\mathscr{A}} = P_{\text{param}}^T\widetilde{M}P_{\text{param}}$ and $\overline{\mathscr{B}} = 2P_{\text{param}}^T\widetilde{f}$, we see immediately that minimizing $J(U)$ is equivalent to minimizing the QUBO

$$\widehat{\Theta}^T\overline{\mathscr{A}}\widehat{\Theta} + \overline{\mathscr{B}}^T\widehat{\Theta}, \qquad \widehat{\Theta} \in \{0,1\}^{rmL}.$$

Remark 1 *In original proposition (see Section III), NmL binary variables are required. With a basis expansion of order r, this reduces to rmL binary variables. Thus, if $rmL \leq 64$, the QUBO defined by Theorem 2 fits current D-Wave AQC devices.*

Remark 2 *If $r = 1$, then $u_k = \alpha_1\phi_1(k)$ is a time-scaled version of a single m-vector α_1. In particular, if $\phi_1(k) \equiv 1$ for all k, then all time steps share the same m-vector α_1. In general, increasing r allows for better approximation of the control law, but also increases the product rmL. In practice, one picks the smallest r that captures the dominant time-variation while respecting $rmL \leq 64$.*

C. State Variables are basis functions

The classical optimal control theory guarantees that (sub)optimal control law is a *state-feedback* of the form $u_k^* = K^* x_k$ with $K^* \in \mathbb{R}^{m\times n}$ being the optimal gain obtained from solving a Discrete Algebraic Riccati Equation (DARE) [10]. The latter being is *time-invariant* and linear in the current state. This inspires the use the system state variables themselves as basis functions.

We posit that the control has the form $u_k = Kx_k$, with $K \in \mathbb{R}^{m\times n}$ and we propose each coordinate of x_k (at time k) as a "basis function" $\phi_\ell(k)$. Concretely, since $x_k = \begin{bmatrix} x_k^{(1)} & x_k^{(2)} & \cdots & x_k^{(n)} \end{bmatrix}^\top$, we treat the scalar $\phi_\ell(k) = x_k^{(\ell)}$ as the ℓ-th basis function. Then, the m-dimensional control is

$$u_k = Kx_k = \sum_{\ell=1}^{n} x_k^{(\ell)} \alpha_\ell, \quad \alpha_\ell \in \mathbb{R}^m,\ \ell = 1,\ldots,n,$$

where $\alpha_\ell^{(i)} = K_{i,\ell}$, is the ℓ-th column of K. By picking $n \ll N$, we shrink the continuous-decision dimension from Nm to nm. After binary-encoding those nm real coefficients with L bits each, the QUBO uses only nmL binary decision variables, independent of N. Each real scalar $K_{i,\ell}$ is represented in fixed-point form using L bits:

$$K_{i,\ell} = \sum_{b=0}^{L-1} 2^{-b} z_{i,\ell}^{(b)}, \quad z_{i,\ell}^{(b)} \in \{0,1\}, \quad i = 1,\ldots,m,\ \ell = 1,\ldots,n. \tag{35}$$

Collecting all nm entries into $\mathrm{vec}(K) \in \mathbb{R}^{nm}$, we build an offset $a \in \mathbb{R}^{nm}$ and precision matrix $P \in \mathbb{R}^{(nm)\times(nmL)}$ so that $\mathrm{vec}(K) = a + Pz, \quad z \in \{0,1\}^{nmL}$, leading to total number of binary (decision) variables as $n\,m\,L$ instead of NmL.

Remark 3 *Given optimal control law is exactly $u_k = K^\star x_k$, then setting $\phi_\ell(k) = x_k^{(\ell)}$, $\alpha_\ell = K_{\cdot,\ell}$ incurs zero basis-approximation error by construction. The only discrepancy arises from finite-bit quantization. We typically scales each state coordinate to $[-1,1]$ or imposes box constraints so that multiplications by bits 2^{-b} remain numerically well-behaved.*

V. Time Complexity analysis

For the classical least-squares solver, building cost matrices and solving via dense inversion of size Nm (see the steps (8),(9)) scales as $O((Nm)^3) \approx O(N^3)$. By contrast, fixing r,m,L so that $rmL \le 64$, the QUBO pipeline constructs the reduced Hessian in $O(N)$, executes the quantum anneal in $O(1)$, and reconstructs trajectories in $O(N^2 n)$, greatly outperforming $O(N^3)$.

VI. Simulation and results

We consider the controllable multi input multi output (MIMO) system (1) with $A = \mathrm{diag}(0.9, 0.8, 0.7) \in \mathbb{R}^{3\times3}$ and $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \in \mathbb{R}^{3\times2}$ so that $n = 3$, $m = 2$, and $\mathrm{rank}([\,B,\ AB,\ A^2B\,]) = 3$. We fix $Q = 100I_3$, $R = 1I_2$, $Q_f = 100I_3$.The initial state is $x_0 = [1,\ 1,\ 1]^\top$. State trajectory X and control U follow from $X = Hx_0 + GU^*$ (see Section II-A). The state and control trajectories for data points $N = 100$ is shown in Fig. 1 wherein all system states as well as control inputs converge to the system's zero confirming that the expected optimal performance is obtained.

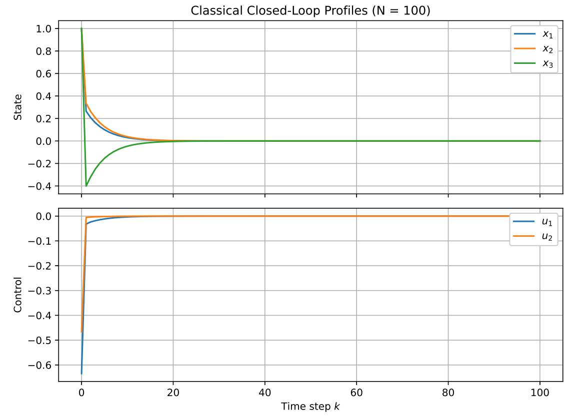

Fig. 1. closed loop action under classical least square based optimal control

For the QUBO approach, we use the state variables as basis functions, so that $r = n = 3$ functions. Each of the $rm = 6$ coefficients is encoded with $L = 8$ bits on $[-3,3]$, so that $rmL = 3\times2\times8 = 48 \le 64$. We sweep $N \in \{100, 300, 600, 1024, 3200, 6400, 12800,$ $25600, 30000\}$ on a computer with Intel Xeon W-2245: 8 cores, 16 threads, 3.9 GHz base. For each N, we time the solve step.

Preprocessing (matrix assembly) is identical for both methods and costs $O(N^2)$, so we exclude it and report only core solver times—dense inversion for LQR vs. quantum anneal—highlighting substantial performance gains. Further, for QUBO control problem, we compute the classical trajectory $X_{\mathrm{cl}}(N)$ and QUBO trajectory $X_{\mathrm{qubo}}(N)$, and report Relative Error $= \|X_{\mathrm{qubo}} - X_{\mathrm{cl}}\|_2$ and QUBO solve time (500 reads). For QUBO problem simulation, we use the *neal* library which is D-Wave's open-source, pure-Python implementation of a simulated-annealing sampler for QUBO and Ising problems.

The solve times for both LS approach and QUBO formulation based approach is shown in Fig. 2. In the plot, for small N (e.g. $N = 500, 1000$), both methods solve almost instantaneously. As N increases to a few thousand, the classical approach begins to exhibit its cubic growth: at $N = 6000$, the least-squares time is already on the order of several seconds, and by $N = 12\,000$ it exceeds tens of seconds. Meanwhile, the QUBO-based solve time remains nearly flat—on the order of milliseconds—because its quantum annealing step is effectively constant (hardware-limited to 64 binary (decision) variable), and its pre-processing costs grow only like $O(N^2)$. Once N reaches $25\,000$ and $30\,000$, the classical solver's time soars to hundreds and then over a thousand seconds, whereas the QUBO time stays under a few seconds. This dramatic divergence underscores the predicted asymptotic crossover: for large horizons, the $O(N^2)$-like QUBO approach (with fixed r,m,L) dramatically outperforms the $O(N^3)$ classical method,

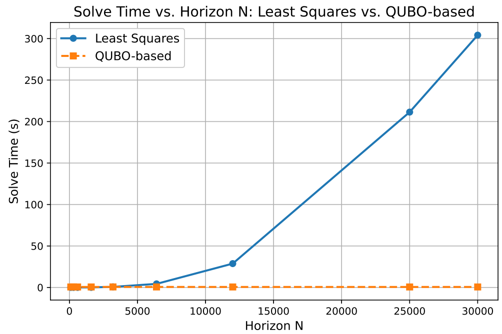

Fig. 2. Solve times for classical least square based formulation and QUBO based control

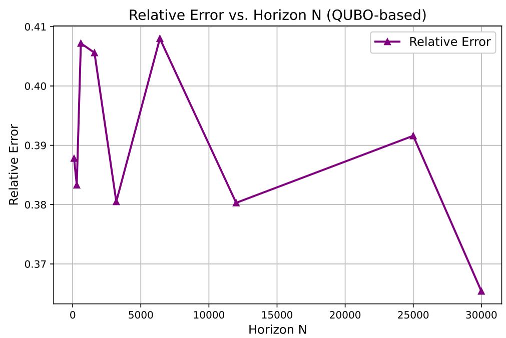

Fig. 3. Relative error between the QUBO-recovered trajectory and the exact least-squares trajectory

validating our theoretical scaling claims.

Fig. 3 plots the relative error between the QUBO-recovered trajectory and the exact least-squares trajectory as the horizon N grows from a few hundred to 3×10^4. Despite this ten-fold increase, the error remains tightly clustered around 0.38–0.41. Two main factors explain this relative error. First, the state-feedback basis $\phi_\ell(k) = x_k^{(\ell)}$ spans the true control law yet suffers finite-precision noise. Second, encoding rm coefficients in $L=8$ bits over $[a,b]$ causes quantization error; rounding yields discrepancy $\|\Theta^* - (a + P\widehat{\Theta})\|$. In an ideal scenario, raising L (finer bit-grid) or adaptively scaling $[a,b]$ around expected coefficient magnitudes would reduce quantization error toward zero. The D-Wave 64-qubit limit ($rmL \le 64$) prevents increasing L or r. Future annealers with more qubits will support finer bit-grids or larger bases, reducing quantization error and enhancing accuracy for larger MIMO systems.

VII. Conclusion and Future Directions

In this work we have introduced a novel QUBO-based reformulation of finite-horizon linear optimal control by exploiting the block-Toeplitz least-squares structure. We show how any Toeplitz-matrix least-squares problem can be converted into a binary-quadratic form whose size depends only on the number of basis coefficients times the chosen bit-precision. By selecting a small basis dimension r (e.g., using state-feedback or spline bases) and encoding each coefficient with L bits, the total number of binary (decision) variable becomes rmL, which can be kept below the current D-Wave limit of 64. Empirical scalability studies confirm that for horizons $N \ge 10^3$, the QUBO pipeline comprising of classical overhead plus most importantly, a hardware-constant anneal time outperforms a classical least square solve. By capping the binary-variable count at 64, today's quantum annealers can handle moderately sized MIMO examples (e.g., $n=3, m=2$) with acceptable error; as annealers scale to hundreds of binary (decision) variable, larger systems and finer precision will be feasible, reducing quantization error. It is noted that linear optimal control also is often addressed by solving a sequence of Algebraic Riccati Equations (AREs), yielding efficient feedback gains that scale linearly with data. However, this is not the point of investigation in this work. Future work will extend these ideas to Riccati-based infinite-horizon control, exploring how discrete AREs may be embedded into QUBO form. As quantum hardware evolves, QUBO-driven optimal control promises to become a competitive, scalable alternative for real-time, high-dimensional control tasks.

VIII. Acknowledgment

The first author would like to thank the good Costa Brava beaches, sangria and Ludwig Göransson for the music in the movie Oppenheimer.

References

[1] Tameem Albash and Daniel A Lidar. "Adiabatic quantum computation". In: *Reviews of Modern Physics* 90.1 (2018), p. 015002.

[2] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[3] Bernard d'Espagnat. *Conceptual foundations of quantum mechanics*. CRC Press, 2018.

[4] Prasanna Date, Robert Patton, Catherine Schuman, and Thomas Potok. "Efficiently embedding QUBO problems on adiabatic quantum computers". In: *Quantum Information Processing* 18 (2019), pp. 1–31.

[5] Prasanna Date and Thomas Potok. "Adiabatic quantum linear regression". In: *Scientific reports* 11.1 (2021), p. 21905.

[6] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. "Quantum computation by adiabatic evolution". In: *arXiv preprint quant-ph/0001106* (2000).

[7] Gene H Golub and Charles F Van Loan. "Matrix computations, 4th". In: *Johns Hopkins* (2013).

[8] Mayank Shekhar Jha, Didier Theilliol, and Philippe Weber. "Model-free optimal tracking over finite horizon using adaptive dynamic programming". In: *Optimal Control Applications and Methods* ().

[9] Andrew Lucas. "Ising formulations of many NP problems". In: *Frontiers in physics* 2 (2014), p. 5.

[10] Robert F Stengel. *Optimal control and estimation*. Courier Corporation, 1994.

[11] Dong Jun Woun, Kathleen Hamilton, Eduardo A Coello Perez, Mayanka Chandra Shekhar, Francisco Rios, John Gounley, In-Saeng Suh, Travis Humble, Georgia Tourassi, et al. "Adiabatic quantum support vector machines". In: *Quantum Machine Intelligence* 7.1 (2025), pp. 1–14.