Abstract—For various typical cases and situations where the formulation results in an optimal control problem, the linear quadratic regulator (LQR) approach and its variants continue to be highly attractive. In certain scenarios, it can happen that some prescribed structural constraints on the gain matrix would arise. Consequently then, the algebraic Riccati equation (ARE) is no longer applicable in a straightforward way to obtain the optimal solution. This work presents a rather effective alternative optimization approach based on gradient projection. The utilized gradient is obtained through a data-driven methodology, and then projected onto applicable constrained hyperplanes. Essentially, this projection gradient determines a direction of progression and computation for the gain matrix update with a decreasing functional cost; and then the gain matrix is further refined in an iterative framework. With this formulation, a data-driven optimization algorithm is summarized for controller synthesis with structural constraints. This data-driven approach has the key advantage that it avoids the necessity of precise modeling which is always required in the classical model-based counterpart; and thus the approach can additionally accommodate various model uncertainties. Illustrative examples are also provided in the work to validate the theoretical results.

Index Terms—Data-driven control, gradient descent, linear quadratic regulator (LQR), optimal control, structural constraints.

I. INTRODUCTION

IN MANY control systems cases that are commonly encountered, it is certainly evident that the methodology of optimal control is highly applicable and effective. As is also commonly known, a specific case of the optimal control problem methodology which is often and regularly utilized is the linear quadratic regulator (LQR) approach. In the LQR problem being considered, it is possible to compose an objective function which is defined as the sum of quadratic terms with respective weightings on state variables and control variables. Notably, the LQR problem admits the optimal solution with several properties in terms of optimality and robustness, and it is straightforward to determine the optimal solution by solving the well-known algebraic Riccati equation (ARE) [1].

With the pervasive power of optimal control, it is noteworthy that a class of controller optimization problems can essentially be formulated as an equivalent LQR problem, but with additional structural constraints imposed on the gain matrix [2]. Most of these structural constraints come from the zero elements in the controller gain matrix, which frequently appear in the context of decentralized control, where the off-block diagonal elements of the composite controller gain are restricted to be zero in the decentralized control architecture. These can also arise from some application-specific factors, such as controller structural restrictions in view of the complexity in measurement and feedback [3]. As is known, the ARE always results in an optimal solution when there are no additional structural constraints (thus known as a full matrix) for the LQR problem. But if certain structural constraints are imposed on the gain matrix, the ARE typically cannot be straightforwardly used; in which case the optimal solution then cannot be easily obtained by similarly straightforward analytical means [4]. Indeed some of these structural constraints typically lead to the NP-hardness of the optimization problem [5], [6], [7], [8].

To address the structural constraints of the controller, some direct methods in optimization can be used [9]. For example, the original optimal control problem can be converted to a nonlinear programming problem [10], and then various approaches can be used, such as sequential quadratic programming, interior point method, etc. In [11] and [12], a gradient projection method is presented to solve the constrained LQR problems with a decentralized gain matrix, which exhibits good properties, such as the monotonic decreasing of the cost, the convergence of
the algorithm, etc. Inspired by these works, linear quadratic constrained optimization is generalized to solve the problems with multiple equality constraints of other types, which is further used for controller synthesis in mechatronic design problems [13]; and the above properties can also be generalized in such settings. In [14] and [15], the structural constraints imposed on the controller can be addressed in an alternate parameter space. In [16], a model-free primal–dual $Q$-learning algorithm is proposed to recover the optimal policy for LQR design, and it is claimed that the main results can be extended in several directions, such as input-constrained optimal control design problems. In [17], an indirect method is presented for solving optimal control problems with state constraints. Note that this approach is based on the application of a refined version of the maximum principle, where the continuity of the corresponding Lagrange multiplier is ensured under certain conditions. There have also been significant progress and success reported in approaches to use the gradient descent method to solve the optimization problem if the gradient can be calculated effectively [18]. Furthermore, the methodology of a projection operator is frequently used to deal with the constraints [19], [20]. These are potentially very attractive and possibly highly effective methodologies; though at this stage, it is certainly unclear from the available existing reported works on how to do the projection on a specific type of constrained hyperplane considering the controller structure. Also, an additional matter of crucial importance is that the determination of the gradient relies on the mathematical model. In some scenarios when the model is inaccurate (such as when there exist significant model uncertainties) or rather difficult to be identified, the model-based optimization approach has to contend with this barrier to provide the “true” optimal solution. Although some robust optimization techniques have been employed to ensure guaranteed robustness toward model uncertainties, it nevertheless still sacrifices optimality to a certain extent [21], [22]. To address such a barrier, there have been some rather promising preliminary studies which utilize the methodology of data-driven approaches in control system synthesis problems [23]. For instance, the iterative feedback tuning (IFT) [24], [25], [26] is a well-known data-driven method for controller optimization and has a wide application in automation. Some representative reinforcement learning algorithms have also been proposed to solve optimal control problems in a model-free framework, such as $Q$-learning [27], [28], [29], adaptive dynamic programming [30], [31], [32], etc. Despite the generality of reinforcement learning frameworks, they are yet to directly handle structural constraints as required by the control system. Also, in terms of the adaptive dynamic programming algorithm, it simplifies a decision by breaking it down into a sequence of decision steps over time. In [33], a policy iteration approach is presented to solve the LQR problem without the knowledge of the internal dynamics (i.e., the state matrix), which is summarized as an algorithm that solves the ARE effectively without knowing a certain part of the system. In [34], an adaptive optimal control scheme with guaranteed Lyapunov stability is presented; and with a policy iteration algorithm based on persistent excitation, the nonlinear ARE is solved assuming unknown system dynamics. Nevertheless in these cases, it is not straightforward from the viewpoint of structural constraints as required by certain control systems. At this current state of these works, it still leaves an important and open question on how to properly solve the linear quadratic optimization problem with structural constraints through a data-driven approach.

In this work, a data-driven optimization algorithm is developed to solve the LQR problem with structural constraints. The gradient of the objective function with respect to the gain matrix is obtained from experiments by injecting an impulse signal to the system. Experimental procedures are given and discussed in terms of the feasibility of practical implementation. The main contributions of this article are essentially encapsulated thus as follows. 1) Through a gradient projection method, the LQR problem with different structural constraints can be solved in an iterative framework with guaranteed convergence. 2) The determination of the gradient of the objective function with respect to the gain matrix is formulated and obtained through a data-driven approach, which demonstrates superiority over the model-based approach in situations of significant model uncertainties. 3) The projection gradient results in terms of zero-element constraints are given with rigorous mathematical derivations. Effectively then, this work further showcases the positive and promising trend to use the data-driven approach as a realistic and effective alternative methodology to achieve optimal performance for industrial control systems.

The remainder of this article is organized as follows. In Section II, the problem statement of the LQR problem with structural constraints is provided. Next, in Section III, the data-driven approach utilized here is presented to derive the gradient estimation of the objective function with respect to the gain matrix. Subsequently, the gradient projection results with zero-element constraints are presented in Section IV. Section V then presents the data-driven optimization algorithm with some of its important properties. In Section VI, numerical examples are given to demonstrate the applicability and effectiveness of the methodology of the proposed algorithm. Finally, pertinent conclusion is drawn in Section VII.

II. PROBLEM STATEMENT

In the usual nomenclature, consider the linear time-invariant (LTI) system

$$\dot{x} = Ax + Bu$$

(1)

with $x(0) = x_0$, where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input vector, $A \in \mathbb{R}^{n \times n}$ is the state matrix, and $B \in \mathbb{R}^{n \times m}$ is the input matrix. As is well known, the objective of the LQR problem is to compute a static state-feedback controller

$$u(K) = Kx$$

(2)

with $K \in \mathbb{R}^{n \times n}$, which stabilizes the closed-loop system and minimizes the objective function

$$J(K) = \int_0^\infty \frac{1}{2} (x^T Q x + u(K)^T R u(K)) \, dt$$

(3)
where \( Q \geq 0 \) and \( R > 0 \). \((A, B)\) is assumed to be stabilizable, \( (A, \sqrt{Q}) \) is assumed to be detectable. For the sake of simplicity, and also aligning with the standard practice in the LQR formulation, we ignore the parameter \( K \) in the expression \( u(K) \) in the following text. In this work, we suppose that the gain matrix is under structural constraints, which is expressed by \( K \in \Phi \), and \( \Phi \) represents the set consisting of all the gain matrices that satisfy the prescribed structural constraints. In particular, the rather commonly occurring case of zero-element constraints is to be considered in this work.

If there is a single constraint imposed on \( K \), the optimization problem can be expressed in the following form:

\[
\min \quad J(K)
\]

subject to \( C(K) = 0 \)

where \( C(K) \) is a linear function of \( K \) with \( C(0) = 0 \). If multiple constraints are imposed on \( K \), the constraints can be expressed as \( C_i(K) = 0 \quad \forall i = 1, 2, \ldots, N \), where \( C_i(K) \) is a linear function of \( K \) with \( C_i(0) = 0 \), and \( N \) is the number of structural constraints.

III. Gradient Estimation

Here, an important step in the procedure is to initially obtain the gradient of the objective function with respect to the gain matrix experimentally. To attain this, we first propose, as a conceptual step only, to inject an impulse signal to the system. Then, it can be observed that to obtain the gradient of the objective function with respect to the gain matrix, which is given by

\[
\frac{dJ(K)}{dK} = \int_0^\infty x^T Q \frac{\partial x(K)}{\partial K} dt + \int_0^\infty u^T R \left( x^T + K \frac{\partial x(K)}{\partial K} \right) dt
\]

with

\[
\frac{\partial x(K)}{\partial K} = \left( \frac{\partial x_1(K)}{\partial K} \right)^T \left( \frac{\partial x_2(K)}{\partial K} \right)^T \ldots \left( \frac{\partial x_n(K)}{\partial K} \right)^T \right) \]

\[
\frac{\partial x_i(K)}{\partial K} = (A_+(K)B\delta e_i^T A_+(K)B)^T
\]

where \( \delta \) represents the impulse signal, and \( e_i \in \mathbb{R}^n \), \( \forall i = 1, 2, \ldots, n \) is the standard basis vector (the \( i \)th entry is one while the others are zero).

Proof of Theorem 1: It is straightforward to obtain the gradient of the objective function with respect to the gain matrix, which is given by

\[
\frac{dJ(K)}{dK} = \int_0^\infty x^T Q \frac{\partial x(K)}{\partial K} dt + \int_0^\infty u^T R \frac{\partial u}{\partial K} dt
\]

Then, it can be observed that to obtain the gradient of the objective function with respect to the gain matrix, the gradient of the state vector with respect to the gain matrix must be known, while the state variables and the control input can be easily captured from the experiment.

From Lemma 1, we have

\[
A_+(K + \varepsilon \Delta K) = (sl - A - BK)^{-1}
\]

\[
= (sl - A - BK)^{-1} + (sl - A - BK)^{-1} B\delta \Delta K (sl - A - BK)^{-1} + O(\varepsilon)
\]

\[
= A_+(K) + A_+(K)B\delta \Delta KA_+(K) + O(\varepsilon).
\]

According to [37], \( A_+(K + \varepsilon \Delta K)B\delta \) represents the value of the state vector when an impulse signal is injected to the system with full state measurable. Then, the standard basis vector \( e_i \) can be used to extract the \( i \)th state variable in the state vector. Thus, we have

\[
x_i(K) = e_i^T A_+(K)B\delta
\]

then

\[
x_i(K + \varepsilon \Delta K) = e_i^T A_+(K + \varepsilon \Delta K)B\delta.
\]

Similarly, we have

\[
x_i(K + \varepsilon \Delta K)
\]

\[
= x_i(K) + e_i^T A_+(K)B \delta \Delta KA_+(K) + O(\varepsilon)
\]

\[
= x_i(K) + \Delta x_i(K) B\delta + O(\varepsilon)
\]

\[
= x_i(K) + \varepsilon A_+(K)B\delta e_i^T A_+(K)B\Delta K + O(\varepsilon).
\]
Then, from Lemma 2, it is easy to obtain (6) and (7). This completes the proof of Theorem 1.

Remark 1: According to the definition of the Fréchet derivative [38], the high-order terms do not influence the value of the first-order derivative. Therefore, the equality in (7) is established (instead of a first-order approximation), where the high-order terms can be eliminated directly.

From the viewpoint of practical implementation, injecting the impulse signal to the system is impossible. To tackle this problem, the impulse signal can be replaced by the unit step signal, then (7) is replaced by

$$\frac{\partial x_i(K)}{\partial K} = \left( A_+(K)B \frac{\partial}{\partial t} (\theta \epsilon_i^T A_+(K)B) \right)^T$$

where \( \theta \) represents the unit step signal. Besides, as the numeric derivatives are employed in the gradient estimation, a low-pass filter can be utilized to suppress the effect of noise.

To realize the data-driven implementation of Theorem 1, the expression (7) needs to be derived from the data of experiments. Specifically, two experiments are required to determine the gradient. Experiment 1 aims to obtain \( \partial (\theta \epsilon_i^T A_+(K)B) / \partial t \) (equivalent to \( \partial (\theta e_i^T A_+ B) / \partial t \)), which represents the derivative of the state variables when injecting the unit step signal through the input channel of the system. In this article, the superscripts “(1)”, “(2)”, and “(3)” represent Experiments 1–3, respectively. We have \( \theta e_i^T A_+(K)B = x_i(1) \) \( \forall i = 1, 2, \ldots, n \), and it is easy to see that the values of all the state variables, that is, \( x_1(1), x_2(1), \ldots, x_n(1) \) are required to be measured in Experiment 1. Then, \( \partial (\theta e_i^T A_+(K)B) / \partial t \) can be easily derived by taking the derivatives of \( x_1(1), x_2(1), \ldots, x_n(1) \), and we have \( x_1(1), x_2(1), \ldots, x_n(1) \), respectively.

Experiment 2 realizes the remaining part of (7) and we need to inject \( x_1(1), x_2(1), \ldots, x_n(1) \) through the input channel of the system separately and measure the state variables \( x_{1,1}, x_{1,2}, \ldots, x_{1,n} \) (when injecting \( x_1(1) \)), \( x_{2,1}, x_{2,2}, \ldots, x_{2,n} \) (when injecting \( x_2(1) \)), and \( x_{n,1}, x_{n,2}, \ldots, x_{n,n} \) (when injecting \( x_n(1) \)), respectively. Here, define

$$x_i(2) = \begin{bmatrix} x_{i,1}^{(2)} & x_{i,2}^{(2)} & \cdots & x_{i,n}^{(2)} \end{bmatrix}^T$$

$$x_n(2) = \begin{bmatrix} x_{n,1}^{(2)} & x_{n,2}^{(2)} & \cdots & x_{n,n}^{(2)} \end{bmatrix}^T$$

we have

$$\frac{\partial x_i(K)}{\partial K} = \left( \frac{x_i(2)}{\partial t} \right)^T \quad \forall i = 1, 2, \ldots, n.$$  (15)

Finally, the gradient can be easily determined by (5) and (6).

Next, we develop the results for the case with \( m \geq 2 \).

Theorem 2: For the LTI system (1) in the multi-input case with the static state-feedback controller (2) and the objective function (3), if the impulse signal is injected to the system through each input channel, the gradient of the objective function with respect to the gain matrix is given by

$$\frac{dJ(K)}{dK} = \int_0^\infty \sum_{i=1}^n q_i x_i(K) \frac{\partial x_i(K)}{\partial K} \, dt + \int_0^\infty \sum_{j=1}^m r_j u_j \frac{\partial u_j(K)}{\partial K} \, dt$$

with

$$\frac{\partial x_i(K)}{\partial K} = (A_+(K)B \delta \epsilon_i^T A_+(K)B)^T$$

where \( q_i \) \( \forall i = 1, 2, \ldots, n \) and \( r_j \) \( \forall j = 1, 2, \ldots, m \) represent the entries located at the \( i \)th row and the \( j \)th column of \( Q \) and the \( j \)th row and \( j \)th column of \( R \), respectively. \( \delta \epsilon \in \mathbb{R}^m \) represents a vector of the impulse signals. \( e_i \) \( \in \mathbb{R}^n \) \( \forall i = 1, 2, \ldots, n \) is the standard basis vector, \( k_j \) represents the entry of the matrix \( K \) located at the \( j \)th row and \( i \)th column, and \( \partial x_i(K) / \partial k_{i1}, \ldots, \partial x_i(K) / \partial k_{im} \) are the entries located at the 1st row and 1st column, \( m \)th row and \( n \)th column of \( \partial x_i(K) / \partial K \) in (17).

Proof of Theorem 2: Equation (16) is straightforward through matrix decomposition and the proof of (17) is similar to the one in Theorem 1. Next, we are going to complete the proof of (18).

The system input is given by \( u_j = \sum_{i=1}^n k_{ji} x_i \). Then, we have

$$\frac{\partial u_j(K)}{\partial k_{fg}} = \sum_{i=1}^n \left( \frac{\partial k_{ji}}{\partial k_{fg}} x_i + k_{ji} \frac{\partial x_i(K)}{\partial k_{fg}} \right)$$

where \( e_j \in \mathbb{R}^m \) \( \forall j = 1, 2, \ldots, m \) is the standard basis vector. Because \( \partial K / \partial k_{fg} \) is a matrix with one entry as one located at the \( f \)th row and \( g \)th column while the other entries are zero, (19) can be further simplified as

$$\frac{\partial u_j(K)}{\partial k_{fg}} = \begin{cases} e_j^T x + \sum_{i=1}^n k_{ji} \frac{\partial x_i(K)}{\partial k_{fg}} & \text{when } j = f \\ \sum_{i=1}^n k_{ji} \frac{\partial x_i(K)}{\partial k_{fg}} & \text{when } j \neq f \end{cases}$$

where \( e_g \in \mathbb{R}^n \) is a standard basis vector in which the \( g \)th entry is one and all the other entries are zero. Arrange (20) into matrix form, then we have (18). This completes the proof of Theorem 2.

Remark 2: Here, it is pertinent to mention that we use the notation that is commonly utilized in the work of the IFT
approach [24], [25]; and thus the operator $p = d/dt$, when utilized, denotes straightforwardly the operation $px(t) = \dot{x}(t)$. Further thus, also straightforwardly, for a stable rational transfer function $H(s)$, the notation $x_1(t) = H(p)x_2(t)$ denotes the output signal $x_1(t)$ when an input signal $x_2(t)$ is passed through the system with the transfer function $H(s)$, which is equivalent to the expression $X_1(s) = H(s)X_2(s)$ in the transform domain (apart, of course, from an inconsequential term arising from initial conditions in the time domain which dies away exponentially fast) as long as $H(s)$ [and thus also $H(p)$] is a rational function (in $s$ and $p$, respectively) and Hurwitz in its denominator polynomial. Following then this same standardized notation used commonly in the work of the IFT approach [24], [25], our methodology (which essentially builds upon, and further extends the IFT approach) thus likewise uses this same adopted compact notation appropriately. It compactly combines the time-domain signals and the transform-domain transfer function in the above-mentioned notationally straightforward fashion without requiring further elaborate (and unnecessarily cumbersome) explicit interpretations when deriving the gradient information, for example, in Theorems 1 and 2.

**Remark 3:** Theorems 1 and 2 are valid, given a feedback gain $K$ that stabilizes the system, and the system is operated in the closed loop. In this case, the optimal solution exists. Additionally, Theorem 1 is a particular case of Theorem 2 when the weighting matrices $Q$ and $R$ are diagonal. However, Theorem 1 can also be used when the weighting matrices $Q$ and $R$ are nondiagonal, while Theorem 2 loses this property.

Similarly, the gradient can be determined through two experiments. By replacing the impulse signal with the unit step signal, (17) is replaced by

$$\frac{\partial x_1(K)}{\partial K} = \left(A_+ + K\frac{\partial}{\partial t}(\theta e^T A_+ B)\right)^T$$

(21)

where $\theta \in \mathbb{R}^m$ represents a vector of the unit step signals.

Experiment 1 aims to realize $\partial(\theta e^T A_+ B)/\partial t$. However, in the multi-input case, $\partial(\theta e^T A_+ B)/\partial t$ is a scalar, so $\partial(\theta e^T A_+ B)/\partial t = \partial(\theta e^T A_+ B\theta)/\partial t$. Therefore, transformations on (21) must be made to ensure the feasibility of the experiment. Take the first state variable $x_1$ as an example, we have $e^T = [1\ 0\ \cdots\ 0]$, then

$$\frac{\partial}{\partial t}(\theta e^T A_+ B) = \frac{\partial}{\partial t}[1\ 0]A_+ B\frac{\partial(\theta e^T A_+ B)}{\partial t}$$

(22)

where $1$ and $0$ represent the vector with all entries as one and the vector with all entries as zero, respectively. In this operation, $\partial(\theta e^T A_+ B)/\partial t$ can be realized by injecting the unit step signal to $m$ input channels separately, taking the differentiation and premultiplying a matrix. Equation (22) can be written in the following form:

$$\frac{\partial}{\partial t}(\theta e^T A_+ B) = \left[\begin{array}{c}1 \\ 0\end{array}\right]\dot{x}_1^{(1)}$$

(23)

where $x_1^{(1)} = A_+ B\left[\theta \ 0 \ \cdots\ 0\right]^T$ can be measured with the unit step signal injected to the 1st input channel only, $\dot{x}_1^{(1)} = A_+ B\left[0 \ \cdots\ 0 \ \theta\right]^T$ can be measured with the unit step signal injected to the $m$th input channel only. In a similar way, $\partial(\theta e^T A_+ B)/\partial t \ \forall i = 1, 2, \ldots, n$ can be obtained, and then define

$$\dot{x}_1^{(1)} = \frac{\partial}{\partial t}(\theta e^T A_+ B) = \left[\begin{array}{c}1 \\ 0\end{array}\right]\dot{x}_1^{(1)}$$

(24)

$$\dot{x}_n^{(1)} = \frac{\partial}{\partial t}(\theta e^T A_+ B) = \left[\begin{array}{c}0 \\ 1\end{array}\right]\dot{x}_n^{(1)}.$$
Similarly, all the other state variables can be measured with the same technique as mentioned above.

Experiment 3 is required to measure $x_1, x_2, \ldots, x_n$, this step can be easily done by injecting the unit step signal to all $m$ input channels and take the differentiation of the state variables, that is, $x_\langle 3 \rangle_i = \frac{\partial (e^T_i A + (K B \theta v))}{\partial t} \forall i = 1, 2, \ldots, n$. Then, the gradient can be easily determined by (16) and (18).

Remark 4: Theorems 1 and 2 demonstrate the results with the impulse signal injected to the system. However, as also noted and indicated earlier, it is the unit step signal which will be used in practical implementation (with the additionally developed equivalent analytical relationships). It is also essential to carefully note that in the developments, the measured system state variables and input signals (which are denoted by respective symbols with superscripts “$\langle 1 \rangle$,” “$\langle 2 \rangle$,” and “$\langle 3 \rangle$”) are different from the ones in Theorems 1 and 2 which are annotated differently; and that the methodology utilizes numerical derivatives.

IV. GRADIENT PROJECTION

On the basis of [12], the gradient projection results with zero-element constraints are presented in the following contents. Assume $dJ(K)/dK$ is not null, then we aim to determine the projection gradient matrix $D$ which guarantees the decrease of the objective function with a step size $\alpha$, that is, $J(K - \alpha D) < J(K)$ with $\alpha > 0$. The Frobenius norm of a matrix is denoted by $\| \cdot \|_F$. The optimization problem (4) is equivalent to calculate $D$ such that the Euclidean distance between $D$ and $dJ(K)/dK$ is minimized. Therefore, for the problem with a single constraint, it is straightforward that the optimization problem (4) can be solved by a gradient descent approach, where the constrained gradient matrix can be determined by

$$\min_D \frac{1}{2} \| \frac{dJ(K)}{dK} - D \|_F^2$$

subject to $C(D) = 0$. (30)

The dual problem of (30) is given by

$$\max_\Lambda \min_D \left( \frac{1}{2} \left\| \frac{dJ(K)}{dK} - D \right\|_F^2 + \text{Tr}(\Lambda^T C(D)) \right)$$

(31)

where $\Lambda$ is the Lagrange multiplier with respect to the structural constraint $C(D) = 0$. For the problem with multiple constraints, its dual problem can be expressed similarly, where $\Lambda_i$ is the Lagrange multiplier associated with the structural constraint $C_i(D) = 0$. Remarkably, we assume that the primal problem (30) is strictly feasible. Then, Slater’s condition is satisfied and strong duality always holds. Thus, the optimal solution can be determined by solving the dual problem (31).

In this work, we consider the structural constraints where some elements in the controller gain matrix are zero, which can be represented by $C(D) = GH$ (for a single constraint) or $C(D) = G_i H_i \forall i = 1, 2, \ldots, N$ (for multiple constraints), respectively. Remarkably, $G$, $G_i$, and $G_j$ are full row rank matrices and $H$, $H_i$, and $H_j$ are full-column rank matrices.

| Experiments | Sub-experiments | Input signals | Measured state variables |
|-------------|-----------------|---------------|-------------------------|
| Single-input case |
| Experiment 1 | - | Unit step signal | $x_\langle 1 \rangle_1, x_\langle 1 \rangle_2, \ldots, x_\langle 1 \rangle_n$ |
| Experiment 2 | Sub-experiment 2.1 | $\dot{x}_1^{\langle 1 \rangle}$ | $x_\langle 1 \rangle_1, x_\langle 1 \rangle_2, \ldots, x_\langle 1 \rangle_n$ |
| | - | $\vdots$ | $\vdots$ |
| | Sub-experiment 2.n | $\dot{x}_n^{\langle 1 \rangle}$ | $x_\langle 1 \rangle_1, x_\langle 1 \rangle_2, \ldots, x_\langle 1 \rangle_n$ |
| Multi-input case |
| Experiment 1 | Sub-experiment 1.1 | Unit step signal to the 1st input channel | $x_\langle 1 \rangle_1, x_\langle 1 \rangle_2, \ldots, x_\langle 1 \rangle_n$ |
| | - | $\vdots$ | $\vdots$ |
| | Sub-experiment 1.m | Unit step signal to the $m$th input channel | $x_\langle 1 \rangle_m, x_\langle 1 \rangle_2, \ldots, x_\langle 1 \rangle_n$ |
| Experiment 2 | Sub-experiment 2.1 | $\dot{x}_\langle 1 \rangle_{1\to m}$ to all $m$ input channels | $x_\langle 2 \rangle_1, x_\langle 2 \rangle_2, \ldots, x_\langle 2 \rangle_n$ |
| | - | $\vdots$ | $\vdots$ |
| | Sub-experiment 2.n | $\dot{x}_\langle 1 \rangle_{m\to n}$ to all $m$ input channels | $x_\langle 2 \rangle_m, x_\langle 2 \rangle_2, \ldots, x_\langle 2 \rangle_n$ |
| Experiment 3 | - | Unit step signal to all $m$ input channels | $x_\langle 3 \rangle_1, x_\langle 3 \rangle_2, \ldots, x_\langle 3 \rangle_n$ |

TABLE I

INPUT SIGNALS AND MEASURED STATE VARIABLES REQUIRED IN THE EXPERIMENTAL PROCEDURES

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
These matrices are required to be chosen depending on the specific structure of the controller gain matrix.

Note that in [12], the determination of projected gradient is given considering a single linear constraint, and Lemma 3 restates part of the results in [12] and generalizes the determination of the projection gradient in two scenarios considering the projection onto a single constrained hyperplane and multiple constrained hyperplanes.

**Lemma 3:** The following statements hold.

1) The gradient projected onto a single constrained hyperplane \( C(D) = GDH \) is given by

\[
D = \frac{dJ(K)}{dK} - G^T (GG^T)^{-1} G \frac{dJ(K)}{dK} H (H^T H)^{-1} H^T. \tag{32}
\]

2) The gradient projected onto multiple constrained hyperplanes \( C_i(D) = G_iDH_i \ \forall i = 1, 2, \ldots, N \) is given by

\[
D = \frac{dJ(K)}{dK} - \sum_{i=1}^{N} G_i^T \Lambda_i H_i^T \tag{33}
\]

where \( \Lambda_i \ \forall i = 1, 2, \ldots, N \) can be obtained by solving the following equations:

\[
G_1 \frac{dJ(K)}{dK} H_1 = \sum_{i=1}^{N} G_1 G_i^T \Lambda_i H_i^T H_1
\]

\[
\vdots
\]

\[
G_N \frac{dJ(K)}{dK} H_N = \sum_{i=1}^{N} G_N G_i^T \Lambda_i H_i^T H_N. \tag{34}
\]

**Proof of Lemma 3:** Part of the proof has been provided in [12]. For Statement 1), the necessary and sufficient optimal solution to the dual problem (31) is given by

\[
D = \frac{dJ(K)}{dK} + \frac{\partial}{\partial D} (\text{Tr}(\Lambda^T C(D))) = 0 \tag{35}
\]

which gives

\[
D = \frac{dJ(K)}{dK} - G^T \Lambda H^T. \tag{36}
\]

Since \( GDH = 0 \), it is easy to see that

\[
G \frac{dJ(K)}{dK} H - GG^T \Lambda H^T H = 0. \tag{37}
\]

Therefore, we have

\[
\Lambda = (GG^T)^{-1} G \frac{dJ(K)}{dK} H (H^T H)^{-1}. \tag{38}
\]

Substitute (38) to (36), we have (32). This concludes the proof of Statement 1). The proof of Statement 2) is omitted because it is similar to the proof of Statement 1).

**V. PROPOSED ALGORITHM**

To summarize the above results on solving the controller synthesis problem under structural constraints, the proposed algorithm is shown in Algorithm 1. Also, to ensure the clarity and conciseness of the algorithm, a flowchart is given in Fig. 1.

It is worthwhile to mention that a feasible step size \( \alpha \) should be specified by the user appropriately. Generally, for model-based optimization problems with the use of the gradient descent technique, the step size could be selected by line search. Similarly, for data-driven optimization problems, line search can also be effectively used. The stopping criterion of the optimization algorithm can be implemented in different ways. One way is to define a small \( \epsilon \), such that the optimization algorithm is terminated as soon as \( \|D^r\|_F \leq \epsilon \), which will be discussed later; alternatives of termination condition can be defined in terms of iteration numbers, etc.

It is worthwhile to mention that the algorithm exhibits some important properties, as summarized in Theorem 3, which is restated and generalized from [12]. For clarification of Theorem 3, the iteration number is denoted by \( N^* \). Note that the evaluation of the cost till \( t = \infty \) is not possible, and thus we propose the approximation of this value in a more practical way. Since the cost function value will not be influenced much after the state variables and control input variables converge to zero, it is practical to estimate the cost function value depending on the time instance when state variables and control input variables converge.
Algorithm 1 Data-Driven Linear Quadratic Optimization Algorithm for Controller Synthesis With Structural Constraints

1. **Step 1:** Set the iteration number $i^* = 0$, and initialize the controller $K_0$ such that the closed-loop system is stable.
2. **Step 2:** $i^* = i^* + 1$. For the single-input case, go to Step 3a; for the multi-input case, go to Step 3b.
3. **Step 3a:** Execute Experiment 1: inject the unit step signal to the system, measure the state variables $x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}$ and the system input $u^{(1)}$. Calculate their derivatives $\dot{x}_1^{(1)}, \dot{x}_2^{(1)}, \ldots, \dot{x}_n^{(1)}$ and $\dot{u}^{(1)}$.
4. **Step 4a:** Calculate $\Delta x/K\partial K$ by (14) and (15) and the estimation of $dJ(K^*)/dK$ by (5) and (6). Go to Step 7.
5. **Step 3b:** Execute Experiment 1: inject the unit step signal to the 1st input channel, measure the state variables $x_1^{(1)}, x_1^{(1)}, \ldots, x_1^{(1)}, x_1^{(1)}, \ldots, x_{m,1}^{(1)}$, and the system input $u^{(1)}$; inject the unit step signal to the 2nd input channel, measure the state variables $x_2^{(1)}, x_2^{(1)}, \ldots, x_2^{(1)}, x_2^{(1)}, \ldots, x_{m,2}^{(1)}$; inject the unit step signal to the $m$th input channel, measure the state variables $x_m^{(1)}, x_m^{(1)}, \ldots, x_m^{(1)}, x_m^{(1)}, \ldots, x_{m,m}^{(1)}$.
6. **Step 4b:** Execute Experiment 2: inject $x_i^{(1)}$ to all $m$ input channels, measure the state variables $x_i^{(2)}, x_i^{(2)}, \ldots, x_i^{(2)}, x_i^{(2)}, \ldots, x_{n,m}^{(2)}$; inject $\dot{x}_i^{(2)}$ to all $m$ input channels, measure the state variables $x_i^{(3)}, x_i^{(3)}, \ldots, x_i^{(3)}, x_i^{(3)}$; set the iteration number $i$.
7. **Step 5a:** Calculate $\Delta x/K\partial K$ by (16) and (18), and the estimation of $dJ(K^*)/dK$ by (17) and (18). Go to Step 7.
8. **Step 7:** Determine the projected gradient $D^*$ by Lemma 3.
9. **Step 8:** Update the controller parameters by $K^* = K^* - \alpha D^*$.
10. **Step 9:** If the stopping criterion is reached, terminate the optimization process; otherwise, go back to Step 2.

**Theorem 3:** Given $K_0 \in \mathbb{R}^{m \times n}$ such that $C(K_0) = 0$ and max(Re(eig(A + BK_0))) < 0, the following statements hold.

1. The optimization algorithm ensures local convergence with $\epsilon$-optimality, that is, $\|D\alpha\|_F \leq \epsilon$.
2. $C(K^*) = 0$ for all $\epsilon > 0$.
3. max(Re(eig(A + BK^*))) < 0 for all $\epsilon > 0$.

**Proof of Theorem 3:** Part of the proof has been shown in [12]. For Statement 1, Lemma 2 is used again, and then we have

$$J(K - \alpha D) = J(K) - \alpha \text{Tr} \left( \left( \frac{dJ(K)}{dK} \right)^T D \right) + \mathcal{O}(\epsilon).$$

(39)

Since $D = 0$ is always feasible in (30), we have

$$\left\| \frac{dJ(K)}{dK} - D \right\|_F \leq \left\| \frac{dJ(K)}{dK} \right\|_F$$

(40)

which leads to

$$\text{Tr} \left( \left( \frac{dJ(K)}{dK} \right)^T D \right) \geq \frac{1}{2} \|D\|_F^2.$$ (41)

From (39), it is easy to derive

$$J(K - \alpha D) - J(K) \leq -\frac{1}{2} \alpha \|D\|_F^2$$

(42)

with $\|D\|_F \neq 0$. Hence, if $\|D\|_F \neq 0$, $\exists \alpha > 0$, such that $J(K - \alpha D) < J(K)$ for $\alpha < \alpha^*$. For the case with a single constraint, in view of the necessary optimal condition for (30), we have

$$\frac{dJ(K)}{dK} + G^T \Gamma H^T = 0$$

(43)

with $\Gamma \in \mathcal{R}$, that is, $GKH = 0$. Similarly, for the necessary optimal condition for (31), we have

$$D - \frac{dJ(K)}{dK} + G^T \Lambda H^T = 0$$

(44)

with $\Lambda \in \mathcal{R}$, that is, $GDH = 0$. From (43) and (44), it is easy to observe that if $\|D\|_F = 0$, the original problem is equivalent to the projection problem with $\Gamma = -\Lambda$. Hence, if $\|D\|_F = 0$, KKT conditions of optimization problem (30) are equivalent to those of (31). Similar results apply to the case with multiple constraints, where the algorithm locally converges to $\epsilon$-optimality in finite time. For Statement 2), the projection gradient satisfies the structural constraint in each iteration, that is, $C(D^*) = 0$. Hence, $C(\alpha D^*) = 0$. Since $K^* = K_{r-1} - \alpha D^*$, if $C(K_{r-1}) = 0$, then $C(K^*) = 0$. Therefore, if $\exists K_0 \in \mathbb{R}^{m \times n}$ such that $C(K_0) = 0$, then $C(K^*) = 0$ for $\epsilon > 0$, $\epsilon = 1, 2, \ldots, N^*$. For Statement 3), since any stable closed-loop system corresponds to a finite cost and the cost is monotonically decreasing during iterations according to Statement 1), it is trivial to conclude that the closed-loop stability is preserved as long as any initial control gain stabilizes the closed-loop system. This completes the proof of Theorem 3.
Remark 5: Theoretically, with more data, the performance of the proposed algorithm is better because the estimation of the gradient becomes more accurate. However, more data generally leads to higher computational effort. In this sense, the user needs to balance the performance and computational effort by choosing the amount of data appropriately.

Remark 6: In the proposed algorithm, the gradient of the state vector with respect to the gain matrix can be measured directly. Thus, we need to consider the complexity of calculating the gradient of the objective function with respect to the gain matrix and also the complexity of gradient projection. In the single input case, the gradient of the objective function with respect to the gain matrix is obtained from (5) and (6), and the complexity is $n^4$ (when $Q$ is diagonal, the complexity is reduced to $n^3$). In the multi-input case, the gradient of the objective function with respect to the gain matrix is obtained from (16) and (18). Since it is assumed that $Q$ is diagonal in the multi-input case, it shows that the complexity is $n^2 m + m^2 n = mn(m + n)$. As for the gradient projection, it is straightforward to see that the complexity is $n^2 m + m^2 n = mn(m + n)$ to project onto a single constrained hyperplane, and the complexity is $Nm(m + n)$ to project onto $N$ constrained hyperplanes.

VI. ILLUSTRATIVE EXAMPLE

To illustrate the applicability and effectiveness of the above results, two examples are reproduced from [12] and [39] with modifications. First, Example 1 presents a controller design problem with a single input and a single zero-element constraint. Next, Example 2 is investigated, which presents a controller design problem with multiple inputs and multiple zero-element constraints. Essentially for the control problems in [12] and [39], the system model is assumed to be known, and this is the major difference between our approach and model-based counterpart. Although the models in the two examples are given below, these models are only used for generating the data to execute the proposed data-driven algorithm. It is worth describing further here that in our detailed experimentations, we have indeed proceeded with explorations involving experimental computations with the proposed algorithm to control a suitably larger dimension quadrotor-type system (which has 12 state variables and four control inputs). With the huge amount of data when the dimension of the system (particularly the dimension of control input) increases, and for this aspect which is essentially matters pertaining to practical realization and implementable computations, we can state that with the typical usual computational power which is not of the so-called “super-computing” capability, this huge amount of data practically leads to serious computational burden and, at this stage, cannot yield a practical solution. Nevertheless here, it should be noted that this is an aspect which relates just to matters pertaining to practical realization and implementable computations. In so far as the key theoretical properties of the methodology are concerned, these had been rigorously established in Theorems 1–3. Appropriately here thus, this aligns with Remarks 5 and 6 in the manuscript.

Considering this limitation, and with current practical computational resources, indeed we can only deal with systems with smaller dimensions at this stage.

Example 1: Consider a linear 2nd-order system, where

$$\dot{x} = Ax + Bu, \quad x(0) = x_0$$

and

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -\sqrt{2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Determine an optimal controller

$$K = [k_1 \ k_2]$$

with $k_1 = 0$, such that

$$\min_k \int_0^{\infty} \langle x^T Q x + u^T Ru \rangle \, dt$$

where

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 0.1.$$  

The structural constraint on $K$ can be stated as

$$KH = 0, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}^T.$$  

The controller gain is initialized as

$$K_0 = \begin{bmatrix} 0 & -5 \end{bmatrix},$$

which stabilizes the closed-loop system with a cost of 1.7789. Then, the optimization is executed with a stopping criteria defined as a total of ten iterations. Figs. 2 and 3 illustrate the changes of the cost and the norm during ten iterations. It can be observed that the cost and the norm drop most significantly within the first iteration. For the following iterations, the cost and the norm still decrease, but with smaller amplitudes. After ten iterations, the cost is given by 1.3127 with a norm of $3.429 \times 10^{-7}$, and the optimal controller gain is given by

$$K^* = \begin{bmatrix} 0 & -3.1813 \end{bmatrix}.$$  

With $x_0 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ and the optimal gain matrix, the system response is shown as the solid lines in Fig. 4.

To demonstrate the effectiveness of the proposed method, a comparison is performed using one of the representative model-free linear quadratic optimization methods called the primal–dual $Q$-learning algorithm [16]. For the comparative method, the discount factor, the approximation factor, and the
initial value number are set as $\alpha = 1$, $M = 1000$, and $r = 4$, respectively. Also, the initial value vectors are designed as

$$V = \begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \\ v_4^T \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 10 & 2 \\ 3 & 2 & 0 \\ 5 & 2 & 0 \end{bmatrix}.$$  

The definitions of the above-mentioned parameters and further details of the comparative method can be found in [16]. Considering the structural constraints in this problem, the projection is applied in the comparative method. Besides, as required by the primal–dual Q-learning algorithm, the continuous system is transformed into its corresponding discrete system with a sampling time of 0.01 s. For the sake of fairness, the iteration number is also set as 10, the initial values of the controller gain and the state variables are set as the same as in our proposed method, that is, $K_0 = [0 \ -5]$ and $x_0 = [1 \ 0]^T$. After ten iterations, the cost is given by 1.3251 and the controller gain is given by

$$K^* = [0 \ -3.2311].$$

With this gain matrix, the system response is shown as the dashed lines in Fig. 4. As shown in the figure, it is illustrated that the performance attained by our proposed method and the primal–dual Q-learning algorithm is rather close. According to the comparison of the cost, our approach leads to a slightly lower one than that of the primal–dual Q-learning algorithm, and this illustrates the correctness of our approach.

**Example 2:** Consider the nominal model of a linear 4th-order system, where

$$\dot{x} = Ax + Bu, \quad x(0) = x_0$$

$$u = Kx$$

and

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 12 & 0 & -10 & 0 \\ 0 & 0 & 0 & 1 \\ -12 & 0 & 25 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & -2 \\ 0 & 0 \\ -2 & 5 \end{bmatrix}.$$  

Assume the system is perturbed by model uncertainties (under which is considered to be operated in the actual condition), and all the nonzero elements in the state matrix $A$ vary within $\pm 20\%$ of their nominal values, respectively. In this case, we also consider such a perturbed system, where

$$\dot{x} = \tilde{A}x + Bu, \quad x(0) = x_0$$

$$u = Kx$$

and

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 9.8 & 0 & -9.8 & 0 \\ 0 & 0 & 0 & 1 \\ -9.8 & 0 & 29.4 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & -2 \\ 0 & 0 \\ -2 & 5 \end{bmatrix}.$$  

In terms of the actual operating condition where the system is perturbed, we aim to determine an optimal controller

$$K = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix}, \quad \begin{bmatrix} k_5 \\ k_6 \\ k_7 \\ k_8 \end{bmatrix}$$

with $k_3 = k_4 = k_5 = k_6 = 0$ (which leads to a decentralized control system), such that

$$\min_K \int_0^{\infty} (x^T Q x + u^T R u) dt$$

where $Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ and $R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

The structural constraints on $K$ can be stated as

$$G_1 KH_1 = 0, \quad G_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad H_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$$

$$G_2 KH_2 = 0, \quad G_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$$

$$G_3 KH_3 = 0, \quad G_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad H_3 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

$$G_4 KH_4 = 0, \quad G_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad H_4 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T.$$  

In this example, the initialized controller gain is set as

$$K = \begin{bmatrix} -50 \\ -20 \\ 0 \\ 0 \\ 0 \\ -20 \\ 0 \\ -6 \end{bmatrix},$$

which stabilizes the closed-loop system with a cost of $1.4480 \times 10^3$. Besides, the iteration number is set as 10 and the initial condition is $x_0 = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^T$. In the proposed data-driven method, Figs. 5 and 6 show the changes of the cost and the norm during ten iterations in the optimization. After ten iterations, the cost is given by $8.2534 \times 10^2$ with a norm of $5.6196 \times 10^{-5}$, and the optimal controller gain is given by

$$K^* = \begin{bmatrix} -57.8053 \\ -18.8322 \\ 0 \\ 0 \\ 0 \\ -38.0592 \\ -7.3845 \end{bmatrix}.$$  

With $x_0 = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^T$ and the optimal gain matrix, the system response is shown in Fig. 7.
To further demonstrate the effectiveness and superiority of the proposed method, a model-based constrained linear quadratic optimization method \[12\] is used in this study as the comparative method. It is worthwhile to note that, for the model-based constrained linear quadratic optimization method, the knowledge of the system model is needed when calculating the controller gain. Thus, in this case study, the nominal system is used to calculate the result of the controller gain, which is further applied to the actual perturbed system to observe the performance. With the comparative model-based method, all the settings regarding the initial gain and iteration number remain the same as those in the proposed data-driven method for the sake of fairness. After ten iterations, the result of the controller gain is given by

\[
K^* = \begin{bmatrix}
-54.0897 & -17.8872 & 0 & 0 \\
0 & 0 & -19.2366 & -6.4998
\end{bmatrix}
\]

and the cost is given by $1.2060 \times 10^3$ with a norm of $3.0566 \times 10^{-3}$. With the determined controller gain, the system response is illustrated in Fig. 8.

Comparing the two methods, it can be easily seen that the cost with the proposed data-driven method is significantly lower than the one with the comparative model-based method, and the performance of the proposed data-driven method is obviously better than the one of the comparative model-based method. The reasonable explanation of the phenomena is that the optimum of the result highly relies on the system model with the model-based linear quadratic optimization methods; while with the data-driven approach, the optimal solution is attained only from the data rather than the system model. As a result, the superiority over the model-based approach is demonstrated over the model-based approach in situations of significant model uncertainties.

VII. CONCLUSION

In this work, a data-driven optimization algorithm is developed to solve the LQR problem with structural constraints. The gradient of the objective function with respect to the gain matrix is derived for both the single-input and the multi-input cases, and then it is projected onto the related hyperplanes characterizing the zero-element constraints. In this way, the gain matrix is iteratively updated toward the optimum with structural constraints preserved. The proposed algorithm provides a suitably general and effective methodology for solving the constrained linear quadratic optimization problem without the need for precise modeling. In the work here, the formulation has also been evaluated and verified on numerical examples, where the effectiveness of the theoretical results is successfully and clearly demonstrated. Possible future works include the generalization of the results to nonlinear systems, as well as the problems with cost functions of other types (such as the nonquadratic cost function). And certainly for the pertinent observations that relate to the important aspect of further improvement of the practical scalability of the proposed method, it can be stated here that this is a significant and substantial aspect where for this approach of the proposed methodology, the computations involved in the control of systems with larger dimensions will also be an important matter as part of the future work.

REFERENCES

[1] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Mineola, NY, USA: Dover Publ., 2007.
[2] Y. Cao and W. Ren, “Optimal linear-consensus algorithms: An LQR perspective,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 819–830, Jun. 2010.
[3] Z. Xia, Y. Liu, J. Lu, J. Cao, and L. Rutkowski. “Penalty method for constrained distributed quaternion-variable optimization,” *IEEE Trans. Cybern.*, vol. 51, no. 11, pp. 5631–5636, Nov. 2021.
Xiaocong Li received the B.Eng. degree in electrical engineering from the National University of Singapore, Singapore, in 2013, and the Ph.D. degree in electrical engineering from the NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore in 2017. He is currently a Postdoctoral Fellow with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. He was a Research Scientist with the Mechatronics Group, Singapore Institute of Manufacturing Technology, Agency for Science, Technology and Research, Singapore. His current research interest focuses on making sense of data for controls through learning as well as its applications to robotics and precision motion systems.

Wenxin Wang received the B.Eng. degree in measuring and control technology and instruments from Tianjin University, Tianjin, China, in 2017, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the National University of Singapore, Singapore, in 2018 and 2021, respectively.

He is currently a Research Fellow with the Advanced Robotics Centre, National University of Singapore. His research interests include intelligent control, optimal control, robust control, and motion planning with application to robotic and mechatronic systems.

Masayoshi Tomizuka (Life Fellow, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Keio University, Tokyo, Japan, in 1968 and 1970, respectively, and the Ph.D. degree in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in February 1974. In 1974, he joined the faculty of the Department of Mechanical Engineering, University of California, Berkeley, Berkeley, CA, USA, where he currently holds the Cheryl and John Neerhout, Jr., Distinguished Professorship Chair. His current research interests are optimal and adaptive control, digital control, motion control, and their applications to robotics and vehicles.

Dr. Tomizuka is the recipient of the Best Journal of Dynamic Systems, Measurement, and Control Best Paper Award in 1995 and 2010, the ASME/DSCD Outstanding Investigator Award in 1996, the Charles Russ Richards Memorial Award (ASME) in 1997, the Rufus Oldenburger Medal (ASME) in 2002, the John R. Ragazzini Award (AACC) in 2006, the Richard Bellman Control Heritage Award (AACC) in 2018, the Honda Medal (ASME) in 2019, and the Nathaniel B. Nichols Medal (IFAC) in 2020. He served as the Program Director of the Dynamic Systems and Control Program of the Civil and Mechanical Systems Division of NSF from 2002 to 2004. He served as the Technical Editor for the ASME Journal of Dynamic Systems, Measurement and Control from 1988 to 1993 and the Editor-in-Chief for the IEEE/ASME TRANSACTIONS ON MECHATRONICS from 1997 to 1999. He is a Life Fellow of ASME and a Fellow of the International Federation of Automatic Control (IFAC) and the Society of Manufacturing Engineers.

Tong Heng Lee received the B.A. degree (with First-Class Hons.) in electrical engineering from the University of Cambridge, Cambridge, U.K., in 1980, the M.Eng. degree in electrical engineering from the National University of Singapore (NUS), Singapore, in 1985, and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 1987. He is a Professor with the Department of Electrical and Computer Engineering, NUS, and the NUS Graduate School, NUS NGS. He was a Past Vice-President (Research) of NUS. His research interests are in the areas of adaptive systems, knowledge-based control, intelligent mechatronics, and computational intelligence.

Dr. Lee is currently an Associate Editor of IEEE TRANSACTIONS ON CYBERNETICS, Control Engineering Practice (IFAC), and International Journal of Systems Science (Taylor & Francis, U.K.). In addition, he is a Past Deputy Editor-in-Chief of Mechatronics (IFAC).