

Data-Driven Actuator Allocation for Actuator Redundant Systems

Filippos Fotiadis [✉], *Graduate Student Member, IEEE*, Kyriakos G. Vamvoudakis [✉], *Senior Member, IEEE*, and Zhong-Ping Jiang [✉], *Fellow, IEEE*

Abstract—In this article, we consider the problem of optimally augmenting an actuator redundant system with additional actuators, so that the energy required to meet a given control objective is minimized. We study this actuator selection problem in two distinct cases; first, in the case where the control objective of the system is not known a priori, and second, in the case where the control objective is a linear state-feedback control law. In the latter scenario, knowledge of the system's state and input matrices is required to solve the corresponding actuator selection problem. However, we relax this requirement by exploiting trajectory data gathered from the system, and using them to iteratively approximate the antistabilizing solution of an associated algebraic Riccati equation (ARE). Notably, the proposed iterative procedure is proved to be small-disturbance input-to-state stable even though the ARE associated with it entails no strictly positive-definite constant term; a result that significantly extends prior work. Finally, to further exploit the obtained trajectory data, we show that these can be used to perform online actuator fault detection without knowledge of the system's matrices, and with complexity lower than that of existing methods. Simulations showcase the theoretical findings.

Index Terms—Actuator selection, learning, redundancy, unknown systems.

I. INTRODUCTION

A CONTROL system is defined to be actuator redundant when: 1) the number of actuators composing it is larger than the number of high-level control inputs available for control design; and 2) it is possible to construct any high-level control input using an appropriate choice of actuator commands [1], [2]; a couple of examples include the ADMIRE benchmark aircraft [3] and the Innovative Control Effectors aircraft [4]. Owing to their two aforementioned characteristics, actuator redundant systems

provide the control designer with plenty of degrees of freedom: They allow any desired control objective to be precisely achieved with an appropriate choice of actuator commands, but they also provide the designer with additional flexibility to optimize other, completely unrelated specifications. The problem of optimally allocating these additional degrees of freedom is often referred to as the control allocation (CA) problem, and comprehensive surveys regarding it are given in [5], [6], and [7].

Minimizing the control energy expended in the closed loop is a universal specification in control systems, as it is related to minimizing monetary cost and increasing the longevity of the system's components. In fact, this is one of the objectives included in a variety of common control techniques, such as model predictive control [8] and linear quadratic regulation [9]. Accordingly, minimizing closed-loop actuation energy is a popular CA specification in actuator redundant systems [2], [10], where the additional degrees of freedom provided by the actuation redundancy are exploited so that the desired control objective is met while expending the lowest possible control energy. The solution to this allocation problem is well known and involves a generalized inverse of the actuation matrix [10], though one may need to resort to numerical methods in constrained cases that consider saturation limits [11].

The aforementioned results render clear that actuation commands should be carefully chosen in order to save control energy. However, the careful selection of the actuators themselves is also of crucial importance since an inconsiderate choice of these could also lead to excess energy expenditure. In the optimal CA problem in actuator redundant systems, this energy can be significantly large when the generalized inverse of the actuation matrix is close to being ill-posed [10].

This work draws motivation from the aforementioned facts. Specifically, since the control energy required to meet a given control objective in an actuator redundant system may be unreasonably large, we study the problem of augmenting that system with additional actuators. The actuators should be optimally selected so that the resulting control energy is minimized, while also taking into account the constraint that a given control objective must be simultaneously achieved.¹ To further generalize our results, we perform this actuator selection procedure in a model-free manner by gathering trajectory data from the system and using them to evaluate the control energy that a given set of actuators will yield.

Considerable research effort has been put towards solving the actuator selection problem in a variety of setups. Most

Manuscript received 9 June 2023; accepted 18 October 2023. Date of publication 31 October 2023; date of current version 29 March 2024. This work was supported in part by ARO under Grant W911NF-19-1-0270, in part by NSF under Grant CAREER CPS-1851588, Grant S&AS-1849198, Grant SATC-1801611, Grant CPS-2227185, and Grant CPS-2227153, and in part by the Onassis Foundation-Scholarship ID: F ZQ 064-1/2020-2021. Recommended by Associate Editor S.-i. Azuma. (Corresponding author: Filippos Fotiadis.)

Filippos Fotiadis and Kyriakos G. Vamvoudakis are with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ffotiadis@gatech.edu; kyriakos@gatech.edu).

Zhong-Ping Jiang is with the Control and Networks Lab, Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: zjiang@nyu.edu).

Digital Object Identifier 10.1109/TAC.2023.3328993

¹This is different than classic minimum energy formulations involving Gramians, which select actuators without taking into account the control objective to be achieved.

often, one is usually tasked with choosing the actuators of a system so that either controllability or resilience is maximized. For example, the authors in [12], [13], [14], and [15] designed and optimized Gramian-related objective functions, which quantified the minimum energy required to steer a linear system between two distinct states. On the other hand, in the context of cyber-physical security, the authors in [16], [17], and [18] formulated security-related specifications for actuator/sensor selection in order to enhance resilience towards adversarial inputs. Various other specifications exist as well, such as metrics that quantify linear-quadratic optimality [19], [20], [21]. However, the aforementioned studies do not consider the actuator selection problem in the context of optimal CA. In addition, and most importantly, all these works consider that the model of the system, i.e., the system's state and input matrices, is completely known.

Learning and estimation are often used as a tool to deal with unknown dynamics in control systems [22], with applications that range from robust control [23] to reinforcement learning [24]. Learning also finds applications in the context of actuator selection. For example, the authors in [25] used classification methods to optimize indices of dynamic performance, while [26] used sparse learning techniques and parameter estimation to select the most control efficient actuators to place on a fuselage. However, mathematical guarantees, in the sense of proving that the optimal set of actuators will be learnt, were not provided. In [27], online learning was used to adaptively select the set of actuators optimizing a Gramian-based metric of controllability, without knowledge of the system's dynamics. However, the problem of actuator selection in the context of optimal CA, which requires the employment of different technical tools and is more theoretically challenging, was not considered in [27], which motivates our present article.

Contributions: Different from the aforementioned studies, this article concentrates on the problem of actuator selection in the context of optimal CA, where one is tasked with augmenting a redundant system with additional actuators to decrease the energy expended in the closed loop. Specifically, two actuator selection metrics are proposed which, if optimized, will lead to a decrease in the control energy needed to achieve a given control objective. The first metric applies in scenarios where the control objective in CA is not known beforehand, and quantifies the energy expended in the closed loop across all possible control directions. On the other hand, the second metric applies in scenarios where the control objective in CA is a linear state-feedback control law, and quantifies the energy expended directly across the trajectories of the closed loop.

Unlike most of the existing works in actuator selection (apart from [27]), the actuators minimizing the proposed CA metrics are provably selected without knowledge of the system's dynamics and with convergence guarantees, by gathering input and state data from the system. We achieve this objective of model-free actuator selection by using a modified version of learning-based policy iteration (PI), which finds the antistabilizing solution of an algebraic Riccati equation (ARE) that is directly related to the actuation energy expended in the closed loop. Borrowing the terminology from [28], the proposed PI algorithm is also proved to be small-disturbance input-to-state stable (ISS) even though the ARE associated with it has no strictly positive-definite constant term. This result further extends prior work studying the robustness of PI [28].

Finally, to further exploit the input/state data gathered in the closed loop, we show that the learnt ARE solution can be

used to perform online fault detection in a model-free manner. A similar detection scheme was proposed in [29], where the solution to an ARE was used to detect actuation attacks by using the principle of optimality. However, our work improves the scheme of [29] in two ways: 1) detection can take place without requiring the high-level control input in the closed loop to be the linear-quadratic regulator (LQR) corresponding to the ARE; and 2) no integrals need to be computed in a receding horizon for detection to take place, rather, only instantaneous measurements of the state and the control input, thus significantly reducing computational complexity.

A preliminary version of this work was presented in [30], in which the proofs of the main results were omitted. In addition, unlike the present work, [30] assumed knowledge of the control input matrix, did not study the robustness of its proposed data-driven algorithm with respect to input noise, and did not propose any data-driven actuator fault detection mechanism. Finally, unlike [30], in this article, we also investigate how the proposed approach can be applied to nonlinear systems.

Structure: The rest of this article is organized as follows. Sections II and III provide preliminaries and formulate the actuator selection problem in the context of CA. Section IV provides a metric for actuator selection when the control objective in CA is unknown, as well as when the control objective is a linear state-feedback control law. Section V describes a method to optimize the actuator evaluation metric of Sections IV-C and IV-D without knowledge of the system's dynamics, and Section VI exploits the output of Section V to perform model-free actuator fault detection. Possible extensions as well as limitations are discussed in Section VII, and simulations are performed in Section VIII. Finally, Section IX concludes this article.

Notation: The sets $\mathbb{C}$, $\mathbb{R}$, and $\mathbb{N}$ will denote the set of complex, real, and natural numbers (including zero), respectively. For a finite set S , $\text{card}(S)$ will denote the cardinality of S , while 2^S will denote the power set of S , i.e., the set of all subsets of S . The operators $\otimes$ and $\oplus$ will denote the Kronecker product and sum, respectively. Given a matrix $Z \in \mathbb{R}^{n \times m}$, $\|Z\|_F$ will denote the Frobenius norm of the matrix Z , $\text{vec}(Z) = [z_{1,1} \ z_{2,1} \ \dots \ z_{n,1} \ z_{1,2} \ z_{2,2} \ \dots \ z_{n,m}]^T$ will denote the vectorized form of a matrix, whereas vec^{-1} will perform the inverse of this operation. Moreover, $\mathcal{B}_\delta(Z) = \{X \in \mathbb{R}^{n \times m} : \|X - Z\|_F < \delta\}$, and $[Z]_{i,j}$ will denote the entry in the i th row and j th column of Z . Additionally, if Z is square and symmetric, we define as $\text{vech}(Z) = [z_{1,1} \ z_{2,1} \ \dots \ z_{n,1} \ z_{2,2} \ z_{3,2} \ \dots \ z_{n,n}]^T$ the half vectorized form of Z , and as $\text{vecs}(Z) = [z_{1,1} \ 2z_{2,1} \ \dots \ 2z_{n,1} \ z_{2,2} \ 2z_{3,2} \ \dots \ z_{n,n}]^T$ the scaled half vectorized form of Z . We denote as I_n the identity matrix of order n . For $Y \in \mathbb{R}^{n \times n}$ and $X \in \mathbb{R}^{(2n) \times (2n)}$

symmetric, $\mathcal{H}(X, Y) = \begin{bmatrix} I_n & -Y^T \\ -Y & \end{bmatrix} X \begin{bmatrix} I_n \\ -Y \end{bmatrix}$. For a sequence of matrices $\{X_i\}_{i \in \mathbb{N}}$, $\|X\|_\infty = \sup_{i \in \mathbb{N}} \|X_i\|_F$. For a vector $z \in \mathbb{R}^n$, $\|z\|$ will denote its Euclidean norm, whereas $[z]_i$ will denote its i th entry.

II. SYSTEM DESCRIPTION AND PRELIMINARIES

Consider a continuous-time linear system of the form

$$\dot{x}(t) = Ax(t) + Gv(t), \quad x(0) = x_0, \quad t \geq 0 \quad (1)$$

where $x(t) \in \mathbb{R}^n$ denotes the state with initial condition $x_0 \in \mathbb{R}^n$, $v(t) \in \mathbb{R}^m$ is the control policy (also referred as *high-level control input*), and $A \in \mathbb{R}^{n \times n}$, $G \in \mathbb{R}^{n \times m}$ are the system's state and input matrices, respectively. The control policy $v(\cdot)$ satisfies the following mapping from actuator commands:

$$v(t) = Bu(t), \forall t \geq 0 \quad (2)$$

where $B \in \mathbb{R}^{m \times k}$ is a matrix whose columns comprise the actuators of the system, and $u(t) \in \mathbb{R}^k$ is a vector containing each actuator's commands.

Regarding the system matrices defined above, A comprehensively describes how the system's states, such as the Euler angles and angular velocities in an aerospace system, interact with one another. In addition, the matrix G dictates which subspace of the state space can be actuated. Finally, each column of B represents an actuator of the system, and the numerical values of these columns indicate how efficient an actuator is at actuating a specific direction in the state space.

Throughout this article, it is assumed that the matrices A and G are unknown. In addition, it is assumed that $v(\cdot)$ is a prescribed control policy, in the sense that the actuation commands $u(\cdot)$ should be appropriately chosen to precisely achieve it. For example, $v(\cdot)$ could be a linear state-feedback control law derived from a pole-placement procedure for system (1), or a linear quadratic regulator.

A. Actuator Redundancy

In the specific case that B has full row rank and $k > m$, given a fixed control policy $v(\cdot)$, (2) has an infinite number of solutions with respect to $u(\cdot)$. Any systems with these two properties are called *actuator redundant* [1], [2], because they have more actuators than the minimum needed to construct $v(\cdot)$ from $u(\cdot)$.

Definition 1: [2] The system (1) is called actuator redundant if the mapping (2) from actuator commands to control inputs satisfies $k > m$ and $\text{rank}(B) = m$. $\square$

B. Optimal Control Allocation

Given that the system is actuator redundant, one is able to choose $u(\cdot)$ so that (2) holds, while still having remaining degrees of freedom over further modifying $u(\cdot)$. For example, consider the control input [2]

$$u(t) = B^\dagger v(t) + B_\perp z(t), t \geq 0$$

where $B^\dagger = B^T(BB^T)^{-1}$ is the Moore–Penrose inverse of B , $B_\perp = I_k - B^\dagger B$ is the null-space projection matrix of B , and $z(t) \in \mathbb{R}^k$ is an arbitrary signal. Then, no matter how $z(t)$ is chosen, it always holds that $v(t) = Bu(t)$. It is thus natural to search for an actuator command vector $u(t)$ that not only achieves the control policy requirement (2), but also has some form of minimum energy. This is termed as the minimum energy CA problem.

Let $W = W(B) \in \mathbb{R}^{k \times k}$ be a positive-definite, diagonal matrix that assigns a weight to each actuator, i.e., to each column of B . Then, the minimum energy CA problem can be mathematically described pointwise in time $t \geq 0$ by the following constrained optimization [1]:

$$\begin{aligned} \min_{u(t) \in \mathbb{R}^k} \quad & L(u(t)) = u^T(t)Wu(t) \\ \text{s.t.} \quad & v(t) = Bu(t). \end{aligned} \quad (3)$$

Evidently, the optimization problem (3) requires that the actuator commands $u(\cdot)$ be chosen so that the control objective (2) is met, while using the remaining degrees of freedom to minimize actuation energy. Following [1], the solution to (3) is given by

$$u^*(t) = W^{-1}B^T(BW^{-1}B^T)^{-1}v(t). \quad (4)$$

Note that, owing to Definition 1, the inverse here always exists for actuator redundant systems. Hence, plugging the optimal actuator command (4) in (3) yields the following constrained minimum value of the weighted energy:

$$L_B^*(t) := L(u^*(t)) = v^T(t)(BW^{-1}B^T)^{-1}v(t). \quad (5)$$

III. PROBLEM FORMULATION

A. Data-Based Actuator Selection

The purpose of this work is to bring connections between CA, actuator selection, and learning. Specifically, note that the minimum allocation energy (5) directly depends on the actuator matrix B . Hence, if one can choose the actuators that will be used by the system, then the minimum energy (5) can be further optimized, but this time with respect to B .

Let the matrix B be decomposed as $B = [B_1 \ B_2]$, where $B_1 \in \mathbb{R}^{m \times k_1}$, $B_2 \in \mathbb{R}^{m \times k_2}$, and $k_1 + k_2 = k$. In this decomposition, B_1 represents the part of B that is fixed and comprises actuators that are already in use by the actuator redundant system, and B_2 represents the part of B that is free to be selected. Define

$$B_1 = [\beta_1 \ \beta_2 \ \dots \ \beta_{k_1}],$$

$$B_2 = [b_1 \ b_2 \ \dots \ b_{k_2}]$$

where $\beta_j \in \mathbb{R}^m$, $j = 1, \dots, k_1$ are fixed actuator columns, and $b_i \in \mathbb{R}^m$, $i = 1, \dots, k_2$, are columns each of which corresponds to an actuator to be selected. Let $\mathcal{S} = \{s_1, \dots, s_N\}$ be the set of available actuator columns $s_i \in \mathbb{R}^m$, where $N \geq k_2$. Then, the problem to be solved in this work is to choose the columns of the free matrix B_2 by solving the optimization

$$\begin{aligned} \min_{B \subseteq \mathcal{S}} \quad & f(B), \\ \text{s.t.} \quad & \text{card}(B) = k_2, \\ & B = \{b_1, b_2, \dots, b_{k_2}\} \end{aligned} \quad (6)$$

where $f: 2^{\mathcal{S}} \rightarrow \mathbb{R}$ is a function quantifying the optimality of the set of actuators B with respect to (5), and is to be defined in the following sections based on the characteristics of $v(t)$, $\forall t \geq 0$. The optimization (6) should be solved in a data-driven manner, without knowledge of the system's matrices A and G .

Before proceeding, we require the following assumption, which ensures that the system (1) and (2) is indeed actuator redundant per Definition 1.

Assumption 1: The matrix $B_1 \in \mathbb{R}^{m \times k_1}$ has full row rank, $k_1 \geq m$ and $k_2 > 0$. $\square$

Remark 1: Given that Assumption 1 holds, then we can see that $\text{rank}(B) = m$ and $k > m$ also hold, hence the system is guaranteed to be actuator redundant. On the other hand, it should be noted that, though sufficient, Assumption 1 is not always necessary to ensure actuator redundancy. $\square$

B. Data-Based Actuator Fault Detection

A subsequent purpose of this work is to ensure that the actuators of the system, which are chosen by the data-based

actuator selection procedure formulated in Section III-A, operate ideally and without any faults. This can be done by monitoring data from the system and detecting any potential actuator faults as soon as they appear. However, owing to the lack of knowledge of the system's dynamics, no utilization of the matrices A and G may be used to perform this detection procedure.

To be more specific, suppose that some of the actuators selected for system (1) suffer from a fault. Then, the actual dynamics of (1) would be described by

$$\dot{x}(t) = Ax(t) + Gv_a(t), \quad t \geq 0 \quad (7)$$

where $v_a(t) \in \mathbb{R}^m$ is a high-level input that has been distorted owing to the actuator fault. Specifically

$$v_a(t) = B(u(t) + a(t)), \quad t \geq 0 \quad (8)$$

where $a(t) \in \mathbb{R}^k$ is the signal modeling the actuator fault. In practice, if the i th actuator of the system, for $i = 1, \dots, k$, is faulty, then the i th entry of $a(t)$ is nonzero, otherwise it is zero. In addition, if there are no actuator faults or failures in the system, then $a(t) \equiv 0$. Hence, to ensure that the actuators selected to be used by the system are not faulty, we would like to evaluate whether $a(t) \equiv 0$ holds online, without precise knowledge of either A or G .

IV. FORMULATION OF THE ACTUATOR SELECTION COST FUNCTION

In this section, we design the cost function f of the actuator selection problem (6) so that it quantifies the value of the minimum energy (5) derived through the CA procedure. Two different formulations are considered for this function, depending on whether the control objective $v(\cdot)$ is a priori known or not. At this point, the results will be completely model-based, i.e., we will assume that A and G are known, but this assumption will be relaxed in Section V by exploiting the results of the present section.

A. Actuator Selection With No Information on the Policy

Consider a general setup in which we want to select the actuators of the system to reduce the energy (5), but where the nominal control policy $v(\cdot)$ is neither fixed nor known beforehand. It can be seen that a direct optimization of L_B^* with respect to B is not possible in this scenario, because L_B^* depends on the undetermined value $v(t)$, $\forall t \geq 0$. Nevertheless, an "average" optimization approach can be applied instead, where the actuators are chosen to minimize (5) evenly across all possible directions of the vector $v(t)$ [13], [14]. Particularly, given the structure of (5), one can choose f to be equal to

$$f_e(B) = \text{tr}((BW^{-1}B^T)^{-1}) \quad (9)$$

where we recall that $W = W(B)$ is a function of B (or $\mathcal{B}$) that assigns a weight to each actuator, and is described as

$$W = \text{diag} [w_{\beta_1} \quad \dots \quad w_{\beta_{k_1}} \quad w_{b_1} \quad \dots \quad w_{b_{k_2}}]$$

with $w_s > 0$ being the weight of each actuator $s \in \mathcal{S}$.²

Albeit elegant, the choice (9) for f entails a computational hurdle: as shown in the upcoming Proposition, the matrix

$(BW^{-1}B^T)^{-1}$ is equivalent to an inverse of a weighted controllability Gramian, and there is no known algorithm that can optimize the trace of such matrices without a combinatorial explosion [31]. In that respect, to optimize (9), one needs to evaluate it for all possible realizations of $\mathcal{B} \subseteq \mathcal{S}$ with $\text{card}(\mathcal{B}) = k_2$, which are $\binom{N}{k_2}$ in number.

Proposition 1: Let $E \in \mathbb{R}^{m \times m}$ be the null matrix. Then, $BW^{-1}B^T$ is a weighted controllability Gramian of the control pair (E, B) .

B. Relaxed Actuator Selection With No Information on the Policy

Relaxations are usually considered to tackle the issue of computational complexity. For example, let $E \in \mathbb{R}^{m \times m}$ be a positive-definite matrix. Then, motivated by the inequality $\frac{\text{tr}(E^{-1})}{m} \geq \frac{m}{\text{tr}(E)}$, a common relaxation to the problem of minimizing $\text{tr}(E^{-1})$ is to maximize $\text{tr}(E)$ instead [12], [32], [33]. Accordingly, this relaxation in the present setup would be equivalent to choosing the function f in (6) as

$$f_{\text{er}}(\mathcal{B}) = -\text{tr}(BW^{-1}B^T). \quad (10)$$

Indeed, such a relaxation is sufficient to reduce the computational complexity to reasonable levels. Particularly, we have

$$\begin{aligned} \text{tr}(BW^{-1}B^T) &= \text{tr} \left(\sum_{j=1}^{k_1} \frac{1}{w_{\beta_j}} \beta_j \beta_j^T + \sum_{i=1}^{k_2} \frac{1}{w_{b_i}} b_i b_i^T \right) \\ &= \sum_{j=1}^{k_1} \frac{1}{w_{\beta_j}} \|\beta_j\|^2 + \sum_{i=1}^{k_2} \frac{1}{w_{b_i}} \|b_i\|^2. \end{aligned} \quad (11)$$

As it is evident from the equation above, optimizing $\text{tr}(BW^{-1}B^T)$ is equivalent to sorting the values $w_s^{-1}\|s\|^2$ for all $s \in \mathcal{S}$, and choosing $\mathcal{B}$ so that it contains the k_2 elements of $\mathcal{S}$ with the largest such values. Hence, the complexity of solving (6) with f chosen as in (10) is $\mathcal{O}(N \log N)$. It should be noted here that the first summation term in (11) is constant, since the actuators β_j , $j = 1, \dots, k_1$, are fixed.

C. Actuator Selection With Prior Information on the Control Policy

The cost function (9) of Section IV-A, as well as its relaxed version (10), quantify the energy (5) on average, across all possible directions for $v(\cdot)$ on $\mathbb{R}^m$. This is a satisfying choice when there is no prior knowledge available regarding the control objective $v(\cdot)$, but may be a naive consideration otherwise. Specifically, if $v(\cdot)$ is known beforehand, then substantially more efficient actuators can be selected for the system by optimizing the actuation energy (5) directly over the predicted trajectories of $v(\cdot)$, rather than across any possible direction of $v(\cdot)$ in $\mathbb{R}^m$.

Motivated by this fact, this subsection considers a more specific setup for the actuator selection problem, in which the control objective $v(\cdot)$ in CA is assumed to have an a priori known structure. Particularly, suppose that $v(\cdot)$ is a linear state-feedback control policy of the form

$$v(t) = Kx(t) \quad (12)$$

where $K \in \mathbb{R}^{m \times n}$ is a known constant gain such that the closed-loop matrix $A + GK$ is Hurwitz. This type of feedback

²Note that W is a function of $\mathcal{B}$ and not of $\mathcal{S}$, as the weights present in its diagonal depend on the choice of actuators $\mathcal{B}$.

is widely employed for controlling linear systems like (1) as it encompasses a variety of goals, including pole placement and linear-quadratic regulation. Its design could be the result of either a model-based procedure that exploits knowledge of the system matrices A , G , or of a model-free one based on data and learning [34], [35], [36]. In addition to (12), we also assume that the initial state $x_0 \in \mathbb{R}^n$ has known covariance, given by

$$\mathbb{E}[x_0 x_0^T] = V \quad (13)$$

where $V \succ 0$.

We now show that with the setup considered above, the metric f in (6) can be chosen to give more specific information regarding the CA energy (5) than the metrics (9) and (10). To this end, notice that if we combine (12) with (5), we obtain the following expression for the value of the CA energy:

$$L_B^*(t) = u^{*T}(t)Wu^*(t) = x^T(t)K^T(BW^{-1}B^T)^{-1}Kx(t).$$

Evidently, this expression is much more informative than that of (5), where the unknown evaluation of the control objective $v(\cdot)$ was involved. Therefore, not only can one choose actuators to optimize control energy “on average,” but we can minimize energy directly on an infinite horizon over the trajectories of (1). Specifically, we can choose f as

$$f_p(\mathcal{B}) = \mathbb{E}\left[\int_0^\infty L_B^*(t)dt\right] = \mathbb{E}\left[\int_0^\infty u^{*T}(t)Wu^*(t)dt\right] \quad (14)$$

where the integration is taken over the trajectories of (1).

It could be argued here that the advantage of considering a more informative cost function is diminished by the fact that integrations and expectations are involved in (14), hence rendering the optimization (6) with $f = f_p$ more complex. However, in the following theorem it is shown that (14) can be written in a completely static and expectation-free manner, as a function of the model matrices A and G .

Theorem 1: Consider the system (1) under the control policy (12), and with initial state covariance given by (13). Then

$$f_p(\mathcal{B}) = \text{tr}(QR) = \text{tr}(PV) \quad (15)$$

where $R = K^T(BW^{-1}B^T)^{-1}K$, and $P, Q \in \mathbb{R}^{n \times n}$ are symmetric matrices, with $P \succeq 0$, $Q \succ 0$, satisfying the Lyapunov equations (LEs)

$$(A + GK)^T P + P(A + GK) + R = 0, \quad (16)$$

$$(A + GK)Q + Q(A + GK)^T + V = 0. \quad (17)$$

Remark 2: It is evident that no integrations or expectations are involved in the expression (15) for f_p , despite the way it was defined in (14). Therefore, (15) provides more information than (9), (10) regarding the control energy expended in the closed loop, but without additional complexity. $\square$

Based on Theorem 1, optimizing the cost (14) has complexity similar to that of optimizing (9), owing to the appearance of the matrix inverse $(BW^{-1}B^T)^{-1}$ through R . Fortunately, however, Theorem 1 suggests that the solution of just one LE is required to evaluate f at all points in 2^S . Specifically, if f is chosen as in (14), then its realization is

$$f_p(\mathcal{B}) = \text{tr}(QR) \quad (18)$$

where we can see from (17) that Q is completely independent of $\mathcal{B}$. Therefore, although a brute-force algorithm to solve (6) with

$f = f_p$ may require a large number of iterations, the per-iteration complexity will remain at relatively low levels as it will not involve the iterative solution of the LE (17).

For the same reason, one can also avoid using the alternate form $f(\mathcal{B}) = \text{tr}(PV)$ provided by Theorem 1, because P is the solution of an LE that depends directly on $\mathcal{B}$. Hence, a brute-force algorithm that would evaluate $f(\mathcal{B}) = \text{tr}(PV)$ across all possible points in 2^S would have to solve the LE (16) at each iteration, which would then lead to an increased per-iteration complexity.

D. Relaxed Actuator Selection With Prior Information on the Control Policy

Consider now the scenario that the gain matrix K has full row rank. Then, it can be seen that

$$\begin{aligned} \text{tr}(QR) &= \text{tr}(QK^T(BW^{-1}B^T)^{-1}K) \\ &= \text{tr}((BW^{-1}B^T)^{-\frac{1}{2}}KQK^T(BW^{-1}B^T)^{-\frac{1}{2}}). \end{aligned}$$

To reduce the complexity of optimizing (18), one can employ the same relaxation that was used in Section IV-B; in lieu of minimizing $\text{tr}(QR)$, one can instead minimize

$$\begin{aligned} f_{pr}(\mathcal{B}) &= -\text{tr}\left(\left((BW^{-1}B^T)^{-\frac{1}{2}}KQK^T(BW^{-1}B^T)^{-\frac{1}{2}}\right)^{-1}\right) \\ &= -\text{tr}(BW^{-1}B^T(KQK^T)^{-1}) \\ &= -\text{tr}(W^{-1}B^T(KQK^T)^{-1}B). \end{aligned} \quad (19)$$

The inverse of KQK^T exists owing to Theorem 1 and the full row rank of K . Subsequently, a further analysis of (19) yields

$$\begin{aligned} f_{pr}(\mathcal{B}) &= -\sum_{j=1}^{k_1} \frac{1}{w_{\beta_j}} \beta_j^T (KQK^T)^{-1} \beta_j \\ &\quad - \sum_{i=1}^{k_2} \frac{1}{w_{b_i}} b_i^T (KQK^T)^{-1} b_i \end{aligned} \quad (20)$$

where the first summation term is constant. We can now notice that the employed relaxation once again brings the complexity down to reasonable levels: To minimize f_{pr} , one only needs to sort the values $\frac{1}{w_s} s^T (KQK^T)^{-1} s$ for all $s \in \mathcal{S}$, and choose $\mathcal{B}$ to contain the k_2 elements of $\mathcal{S}$ with the largest such values. Therefore, the complexity of solving (6) with f as in (20) is $\mathcal{O}(N \log N)$, and only one LE of the form (17) needs to be solved.

V. MODEL-FREE COMPUTATION OF THE ACTUATOR SELECTION COST WITH PRIOR INFORMATION

The results of the previous section, and specifically of Sections IV-C/IV-D, were derived in a completely model-based manner: To evaluate the energy required by a particular set of actuators $\mathcal{B}$ towards achieving the control objective (12), the cost functions (18) and (19) had to be calculated and optimized. However, these cost functions directly depend on the system's matrices A and G through the LE solution Q , and hence cannot be evaluated directly in a model-free manner. This section relaxes this requirement of system knowledge: By utilizing state/input data gathered from the system (1), it is shown that the matrices Q and Q^{-1} , and consequently the cost functions (18)

and (19), can be evaluated directly with no knowledge of either A or G .

A. Antistabilizing Solution to an ARE

Existing data-driven methods, such as those in [27] and [34], are able to solve LEs of the form (16) in a model-free manner, but they cannot solve LEs of the form (17) owing to the transpose operators being applied at an inconvenient position. But, as explained in Section IV-C, it is the LE (17) that we are actually interested in solving, first because it is independent of $\mathcal{B}$ and hence, we need to solve only one of it to optimize $f_p(\cdot)$, and second, because its solution can be used to obtain the computationally relaxed cost function $f_{pr}(\cdot)$ in (20). Towards dealing with this issue of the transpose operators of the LE (17) being misplaced, we will use the following idea: Since Q is positive-definite and thus invertible, we can pre- and postmultiply the LE (17) by Q^{-1} and turn it into the following ARE for the plant $A + GK$:

$$(A + GK)^T X + X(A + GK) + XVX = 0 \quad (21)$$

where $X_a = Q^{-1}$ is a solution to (21).

Various methods for solving AREs model-free exist in the literature [34], [37], which are essentially a learning-based formulation of procedures widely known as policy iteration (PI), successive approximations, or Kleinman's algorithm [38], [39], [40]. Accordingly, it is tempting to use these methods to solve (21) in a learning-based manner and compute Q^{-1} without knowledge of A and G . However, there is a pitfall involved in the ARE (21) that would cause the direct implementations of [34] and [37] to fail to find Q^{-1} ; these methods can only find the stabilizing solution to an ARE.

Definition 2: Consider the general form of an ARE

$$\bar{A}^T \bar{X} + \bar{X} \bar{A} + \bar{Q} - \bar{X} \bar{\Sigma} \bar{X} = 0 \quad (22)$$

where $\bar{A}, \bar{X}, \bar{Q}, \bar{\Sigma} \in \mathbb{R}^{n \times n}$, and $\bar{X}, \bar{Q}, \bar{\Sigma}$ are symmetric. Then

- 1) A solution $\bar{X}_s \in \mathbb{R}^{n \times n}$ to the ARE (22) is called stabilizing with respect to $\bar{A}$ if $\bar{A} - \bar{\Sigma} \bar{X}_s$ is Hurwitz.
- 2) A solution $\bar{X}_a \in \mathbb{R}^{n \times n}$ to the ARE (22) is called antistabilizing with respect to $\bar{A}$ if all eigenvalues of $\bar{A} - \bar{\Sigma} \bar{X}_a$ have strictly positive real parts. $\square$

We show next that Q^{-1} is not the stabilizing solution to (21), hence the learning-based PI methods of [34] and [37] would fail to compute it. In fact, we prove that Q^{-1} is the antistabilizing solution of (21).

Lemma 1: Consider the ARE (21), where $X \in \mathbb{R}^{n \times n}$ is the variable to be solved for. Then

- 1) $X_s = 0$ is the stabilizing solution to (21) with respect to $A + GK$.
- 2) $X_a = Q^{-1}$ is the antistabilizing solution to (21) with respect to $A + GK$.

B. PI for the ARE's Antistabilizing Solution

Although Q^{-1} is not the stabilizing solution of (21) with respect to $A + GK$, it can prove handful that it has been characterized as the antistabilizing one. Particularly, due to this property, the matrix Q^{-1} can be shown to be a stabilizing solution to an alternate ARE and with respect to an alternate matrix instead, as we show next.

Lemma 2: $X_a = Q^{-1}$ is the stabilizing solution of

$$-(A + GK)^T X - X(A + GK) - XVX = 0 \quad (23)$$

Algorithm 1: PI to Compute Q^{-1} .

1: Let $i = 0, \epsilon > 0$. Start with a matrix $Y_0 \in \mathbb{R}^{n \times n}$ such that $-(A + GK) - Y_0$ is Hurwitz.

2: **repeat**

3: Compute X_i by solving the LE:

$$-(A + GK + Y_i)^T X_i - X_i(A + GK + Y_i) + Y_i^T V^{-1} Y_i = 0. \quad (24)$$

4: Compute Y_{i+1} as $Y_{i+1} = V X_i$.

5: Set $i = i + 1$.

6: **until** $\|Y_i - Y_{i-1}\|_F < \epsilon$

with respect to $-(A + GK)$.

The matrix Q^{-1} has now been characterized as a stabilizing solution of (23) with respect to $-(A + GK)$. Therefore, we are finally able to use the PI procedure to compute it in an iterative fashion, which is something we could not have done directly on (21) with respect to $A + GK$. Although the PI algorithm is inherently model-based as seen from Algorithm 1, it is the first step towards computing Q^{-1} in a data-based fashion, just as in [34] and [37].

Notice now that the constant term in the ARE (23) is zero, hence the observability assumptions imposed in [34], [37], and [40] for the constant term do not hold. Consequently, in proving convergence of Algorithm 1, we will need to use different arguments from [34], [37], and [40].

Theorem 2: Consider the sequence of matrices $\{X_i\}_{i \in \mathbb{N}}$, $\{Y_i\}_{i \in \mathbb{N}}$ generated by Algorithm 1. Then, the following hold for all $i \in \mathbb{N}$:

- 1) $-(A + GK + Y_i)$ is Hurwitz;
- 2) $Q^{-1} \preceq X_{i+1} \preceq X_i$;
- 3) $\lim_{i \rightarrow \infty} X_i = Q^{-1}$.

Remark 3: Theorem 2 proves convergence of the PI Algorithm 1, even though the ARE (23) involves no constant term. However, this result is owed to the fact that $-(A + GK)$ is specifically an antistable matrix. If, on the other hand, $-(A + GK)$ was not antistable, it would not have been possible to obtain the result of Theorem 2, and the observability assumptions of [34] and [37] for the constant term would have been indispensable. $\square$

Just like all PI-based algorithms, Algorithm 1 requires that the initial matrix Y_0 is stabilizing, i.e., that $-(A + GK) - Y_0$ is Hurwitz. This assumption naturally decreases the degree to which a learning-based PI algorithm can be model-free because it requires some empirical knowledge regarding the state matrix A and the input matrix G . Nevertheless, in the framework considered here, a stabilizing matrix Y_0 can be constructed by just the mere knowledge of a lower bound to the minimum eigenvalue of $A + GK$, as shown next.

Theorem 3: Suppose that $Y_0 = -\alpha I_n$, where $\alpha < \alpha^* = \min_{i \in \{1, \dots, n\}} \text{Re}(\lambda_i(A + GK))$. Then, $-(A + GK) - Y_0$ is Hurwitz.

C. Learning-Based PI for the Antistabilizing Solution

The analysis of the previous section allowed us to successively approximate the antistabilizing solution Q^{-1} of (21) using PI. In view of this result, we now proceed to derive a data-based formulation for Algorithm 1 which does not require knowledge

of the system's matrices A and G . Notice that if we pre- and postmultiply (24) by $x(t)$, where $x(t)$ is the trajectory of (1) for all $t \geq 0$, then

$$\begin{aligned} & -x^T(t)(A^T X_i + X_i A)x(t) - x^T(t)(GK + Y_i)^T X_i x(t) \\ & -x^T(t)X_i(GK + Y_i)x(t) + x^T(t)Y_i^T V^{-1}Y_i x(t) = 0 \end{aligned}$$

or equivalently $\forall t \geq 0$

$$\begin{aligned} & -\frac{d}{dt}(x^T(t)X_i x(t)) + x^T(t)X_i Gv(t) + (Gv(t))^T X_i x(t) \\ & -x^T(t)X_i(GK + Y_i)x(t) - ((GK + Y_i)x(t))^T X_i x(t) \\ & + x^T(t)Y_i^T V^{-1}Y_i x(t) = 0. \end{aligned} \quad (25)$$

Letting, for all $i \in \mathbb{N}$

$$Z_i := X_i G \quad (26)$$

Eq. (25) can be written $\forall t \geq 0$ as

$$\begin{aligned} & -\frac{d}{dt}(x^T(t)X_i x(t)) + 2x^T(t)Z_i(v(t) - Kx(t)) \\ & -x^T(t)Y_i^T X_i x(t) - x^T(t)X_i Y_i x(t) + x^T(t)Y_i^T V^{-1}Y_i x(t) = 0. \end{aligned}$$

Note that in these equations, $v(\cdot)$ could be any signal generating the trajectory data of x , and not necessarily given by (12). Next, if we let $T > 0$ and integrate over $[t, t+T]$, then

$$\begin{aligned} & x^T(t)X_i x(t) - x^T(t+T)X_i x(t+T) \\ & + \int_t^{t+T} (2x^T(\tau)Z_i(v(\tau) - Kx(\tau)) - x^T(\tau)X_i Y_i x(\tau) \\ & - x^T(\tau)Y_i^T X_i x(\tau) + x^T(\tau)Y_i^T V^{-1}Y_i x(\tau)) d\tau = 0. \end{aligned} \quad (27)$$

Each term of this equation can be rewritten using Kronecker algebra as follows:

$$\begin{aligned} & x^T(t)X_i x(t) - x^T(t+T)X_i x(t+T) \\ & = (x^T(t) \otimes x^T(t) - x^T(t+T) \otimes x^T(t+T))\text{vec}(X_i), \\ & 2x^T(\tau)Z_i(v(\tau) - Kx(\tau)) \\ & = 2(v^T(\tau) \otimes x^T(\tau) - (x^T(\tau) \otimes x^T(\tau))(K^T \otimes I_n))\text{vec}(Z_i), \\ & x^T(\tau)X_i Y_i x(\tau) + x^T(\tau)Y_i^T X_i x(\tau) \\ & = (x^T(\tau) \otimes x^T(\tau))(Y_i^T \oplus Y_i^T)\text{vec}(X_i) \\ & x^T(\tau)Y_i^T V^{-1}Y_i x(\tau) = (x^T(\tau) \otimes x^T(\tau))\text{vec}(Y_i^T V^{-1}Y_i). \end{aligned}$$

Hence, (27) is equivalent to

$$\Theta_i(t) \begin{bmatrix} \text{vecs}(X_i) \\ \text{vec}(Z_i) \end{bmatrix} = \Phi_i(t) \quad (28)$$

with

$$\begin{aligned} \Theta_i(t) &= [\text{vech}(\text{vec}^{-1}(\delta_{xx}(t) - J_{xx}(t)(Y_i^T \oplus Y_i^T)), \\ & \quad 2J_{vx}(t) - 2J_{xx}(t)(K^T \otimes I_n)], \\ \Phi_i(t) &= -J_{xx}(t)\text{vec}(Y_i^T V^{-1}Y_i), \\ \delta_{xx}(t) &= x^T(t) \otimes x^T(t) - x^T(t+T) \otimes x^T(t+T), \\ J_{vx}(t) &= \int_t^{t+T} v^T(\tau) \otimes x^T(\tau) d\tau, \end{aligned}$$

Algorithm 2: Learning-Based PI to Compute Q^{-1} .

- 1: Let $i = 0, \epsilon > 0$. Start with a matrix $Y_0 \in \mathbb{R}^{n \times n}$ such that $-(A + GK) - Y_0$ is Hurwitz.
- 2: **repeat**
- 3: Compute X_i and Z_i from (31).
- 4: Compute Y_{i+1} as:

$$Y_{i+1} = V X_i.$$

- 5: Set $i = i + 1$.
 - 6: **until** $\|Y_i - Y_{i-1}\|_F < \epsilon$
-

$$J_{xx}(t) = \int_t^{t+T} x^T(\tau) \otimes x^T(\tau) d\tau.$$

Notice that the matrices A and G are not involved at all in (28). Hence, if data is gathered along the trajectories of (1) in the form of matrices $J_{xx}(t_\kappa)$, $J_{vx}(t_\kappa)$, and $\delta_{xx}(t_\kappa)$, where $t_\kappa > 0$ is a sampling instant for $\kappa \in \{0, \dots, K_0\}$ and $K_0 \in \mathbb{N}$, then the matrices X_i and Z_i can be determined using (28). To this end, denote, with a slight abuse of notation

$$\delta_{xx} = \begin{bmatrix} \delta_{xx}(t_0) \\ \vdots \\ \delta_{xx}(t_{K_0}) \end{bmatrix}, J_{xx} = \begin{bmatrix} J_{xx}(t_0) \\ \vdots \\ J_{xx}(t_{K_0}) \end{bmatrix}, J_{vx} = \begin{bmatrix} J_{vx}(t_0) \\ \vdots \\ J_{vx}(t_{K_0}) \end{bmatrix}$$

and

$$\begin{aligned} \Theta_i &= [\text{vech}(\text{vec}^{-1}(\delta_{xx} - J_{xx}(Y_i^T \oplus Y_i^T)), 2J_{vx} - 2J_{xx}(K^T \otimes I_n)], \\ \Phi_i &= -J_{xx}\text{vec}(Y_i^T V^{-1}Y_i) \end{aligned}$$

where the operator $\text{vech}(\text{vec}^{-1}(\cdot))$ is applied row-wise. Then, stacking (28) yields

$$\Theta_i \begin{bmatrix} \text{vecs}(X_i) \\ \text{vec}(Z_i) \end{bmatrix} = \Phi_i. \quad (29)$$

This equation can be solved for $\text{vecs}(X_i)$ and $\text{vec}(Z_i)$, provided Θ_i has full column rank. This will hold if the measured data is sufficiently rich, a condition described more concretely in the following Lemma.

Lemma 3: If

$$\text{rank}[J_{xx} J_{vx}] = \frac{n(n+1)}{2} + nm \quad (30)$$

then, Θ_i has full column rank for all $i \in \mathbb{N}$.

Remark 4: The rank condition of Lemma 3 is in the spirit of the persistency of excitation condition in adaptive control [36]. Rank conditions of this kind are ubiquitous in virtually all data-driven designs of the literature [41], [42].

Given condition (30), the solution to (29) is given by

$$\begin{bmatrix} \text{vecs}(X_i) \\ \text{vec}(Z_i) \end{bmatrix} = (\Theta_i^T \Theta_i)^{-1} \Theta_i^T \Phi_i. \quad (31)$$

This gives rise to the learning-based PI Algorithm 2 for computing the antistabilizing solution Q^{-1} to (21). Consequently, we are now able to solve the actuator placement problem (6) with cost functions (18) and (19) in a model-free manner.

We summarize the convergence of Algorithm 2 next.

Theorem 4: Let condition (30) hold. Consider the sequence of matrices $\{X_i\}_{i \in \mathbb{N}}$, $\{Y_i\}_{i \in \mathbb{N}}$, $\{Z_i\}_{i \in \mathbb{N}}$ generated by Algorithm 2. Then, for all $i \in \mathbb{N}$

- 1) $-(A + GK + Y_i)$ is Hurwitz;
- 2) $Q^{-1} \preceq X_{i+1} \preceq X_i$;
- 3) $\lim_{i \rightarrow \infty} X_i = Q^{-1}$, $\lim_{i \rightarrow \infty} Z_i = Q^{-1}G$.

Remark 5: According to Theorem 4, Algorithm 2 yields the matrix Q^{-1} as its output by only using data, and without knowledge of the system's matrices A and G . Subsequently, this output can be used to evaluate the actuator selection metrics (18) and (19)—and to solve the actuator selection problem (6)—in a model-free manner. It is worth mentioning that Algorithm 2 estimates only $n(n+1)/2 + nm$ parameters in total, which are strictly fewer than the number of parameters a system identification approach would have to estimate to learn the matrices A and G ($n^2 + nm$ in total). Accordingly, the main operation in Algorithm 2 is the inversion of the $(n(n+1)/2 + nm) \times (n(n+1)/2 + nm)$ data matrix. $\square$

Remark 6: Another advantage of the proposed approach compared to system identification is that, since it directly estimates Q^{-1} , any errors in the estimation of this matrix will manifest themselves in the cost function (19) in a mostly linear manner. On the contrary, any errors a system identification approach would have in estimating the system matrices A, G would manifest themselves in the cost function (19) in a highly nonlinear manner, since Q is derived as the solution of the LE (17). In fact, in the presence of even slight system identification errors, it is uncertain if solving the LE (17) for Q , with estimates of A and G , would be feasible at all. $\square$

D. Robustness in the Presence of Input Noise

In this subsection, we will show that the PI Algorithm 1 (and by extension the learning-based PI Algorithm 2), which was used to compute Q^{-1} , has certain robustness properties with respect to uncertainties. Specifically, we extend the results of [28] to show that it is small-disturbance ISS, with the advancement from [28] being that the associated ARE (23) describing Q^{-1} entails no constant term.

Towards this end, note that (24) can be written equivalently as $\mathcal{H}(\mathcal{G}(X_i), Y_i) = 0$, where for any $X \in \mathbb{R}^{n \times n}$ symmetric

$$\begin{aligned} \mathcal{G}(X) &:= \begin{bmatrix} -(A + GK)^T X - X(A + GK) & X \\ X & V^{-1} \end{bmatrix} \\ &= \begin{bmatrix} [\mathcal{G}(X)]_{xx} & [\mathcal{G}(X)]_{ux}^T \\ [\mathcal{G}(X)]_{ux} & [\mathcal{G}(X)]_{uu} \end{bmatrix}. \end{aligned}$$

Therefore, Algorithm 1 can be equivalently formulated as in the following procedure.

Procedure 1:

- 1) Choose $Y_0 \in \mathbb{R}^{n \times n}$ such that $-(A + GK + Y_0)$ is Hurwitz and let $i = 0$.
- 2) Solve $\mathcal{H}(\mathcal{G}(X_i), Y_i) = 0$ for the symmetric $X_i \in \mathbb{R}^{n \times n}$.
- 3) Set $Y_{i+1} = [\mathcal{G}(X_i)]_{uu}^{-1}[\mathcal{G}(X_i)]_{ux}$, $i = i + 1$ and go to step 2.

From Theorem 2, we know that this procedure guarantees $\lim_{i \rightarrow \infty} X_i = Q^{-1}$. Next, to study the robustness of PI, we follow [28] and consider the perturbed variant of Procedure 1.

Procedure 2:

- 1) Choose $\hat{Y}_0 \in \mathbb{R}^{n \times n}$ such that $-(A + GK + \hat{Y}_0)$ is Hurwitz and let $i = 0$.
- 2) Obtain a symmetric $\hat{\mathcal{G}}_i \in \mathbb{R}^{(2n) \times (2n)}$ as an approximation of $\mathcal{G}(\hat{X}_i)$, where $\hat{X}_i \in \mathbb{R}^{n \times n}$ is the symmetric solution of $\mathcal{H}(\mathcal{G}(\hat{X}_i), \hat{Y}_i) = 0$.
- 3) Set $\hat{Y}_{i+1} = [\hat{\mathcal{G}}_i]_{uu}^{-1}[\hat{\mathcal{G}}_i]_{ux}$, $i = i + 1$ and go to step 2.

The following Lemma shows that Procedure 2 is locally ISS with respect to the induced error $\Delta\mathcal{G}_i := \hat{\mathcal{G}}_i - \mathcal{G}(\hat{X}_i)$:

Lemma 4: [28] For any $\sigma \in (0, 1)$, there exist $\delta_0, \delta_1 > 0$ such that if $\|\Delta\mathcal{G}\|_\infty < \delta_1$ and $\hat{X}_0 \in \mathcal{B}_{\delta_0}(Q^{-1})$

- 1) $[\hat{\mathcal{G}}_i]_{uu}$ is invertible and $-(A + GK + Y_i)$ Hurwitz, $\forall i \in \mathbb{N}$;
- 2) the following local ISS property holds:

$$\|\hat{X}_i - Q^{-1}\|_F \leq \beta_0 \left(\|\hat{X}_0 - Q^{-1}\|_F, i \right) + \gamma_0(\|\Delta\mathcal{G}\|_\infty)$$

where $\beta_0(y, i) = \sigma^i y$, $\gamma_0(y) = c_3 y / (1 - \sigma)$, $y \in \mathbb{R}$ and $c_3(\delta_0) > 0$.

- 3) $\lim_{i \rightarrow \infty} \|\Delta\mathcal{G}_i\|_F = 0$ implies $\lim_{i \rightarrow \infty} \|\hat{X}_i - Q^{-1}\|_F = 0$.

Remark 7: While [28] requires the ARE associated with the PI algorithm to have a positive-definite constant term, it has no such limitation in the statement and proof of Lemma 4. $\square$

In simple words, Lemma 4 states that the matrix sequence $\hat{X}_i$ in Procedure 2 will remain in a neighborhood of Q^{-1} , provided that the uncertainties $\Delta\mathcal{G}_i$ entering the procedure are small. In addition, if these uncertainties vanish, then $\hat{X}_i \rightarrow Q^{-1}$. However, a limitation of Lemma 4 is that it requires initializing $\hat{X}_i$ within a proximity of Q^{-1} . The following Theorem removes this requirement, and extends the results of [28]: it shows that Procedure 2 is small-disturbance ISS, even though the associated ARE (23) entails no constant term.

Theorem 5: For an initial matrix $\hat{Y}_0$ such that $-(A + GK + \hat{Y}_0)$ is Hurwitz and for any $\epsilon > 0$, there exists $\delta_2 > 0$ such that if $\|\Delta\mathcal{G}\|_\infty < \delta_2$

- 1) $[\hat{\mathcal{G}}_i]_{uu}$ is invertible, $-(A + GK + \hat{Y}_i)$ is Hurwitz and $\|\hat{X}_i\|_F < M_0$, $\forall i \in \mathbb{N}$, for some $M_0 > 0$;
- 2) $\limsup_{i \rightarrow \infty} \|\hat{X}_i - Q^{-1}\|_F < \epsilon$;
- 3) $\lim_{i \rightarrow \infty} \|\Delta\mathcal{G}_i\|_F = 0$ implies $\lim_{i \rightarrow \infty} \|\hat{X}_i - Q^{-1}\|_F = 0$.

Theorem 5 shows that Procedure 2 is small-disturbance ISS with respect to the uncertainty induced by $\Delta\mathcal{G}$. Using [28], this result can be used to conclude that Algorithm 2 is also small-disturbance ISS with respect to disturbances in the control input v . Specifically, if, instead of (1), the system dynamics were given by $\dot{x}(t) = Ax(t) + Gv(t) + d(t)$, where $d(t) \in \mathbb{R}^n$ is an unknown additive disturbance, then Algorithm 2 would still converge in an ϵ -neighborhood of Q^{-1} , provided that $\|d(t)\| < \delta$, $\forall t \geq 0$, for some strictly positive $\delta = \delta(\epsilon) > 0$. This result follows by combining Theorem 5 with the proof of [28], Theorem 3].

VI. MODEL-FREE DETECTION OF ACTUATOR FAULTS

Generally, actuators are prone to faults and failures which can cause deterioration of the system's performance. These faults should be detected as soon as possible after their appearance because they can pose a threat to the existence of the physical system by putting closed-loop stability and well-posedness in

jeopardy. Nevertheless, the task of fault detection can be extraordinarily difficult in cases where the matrices A and G are not known a priori, as in the present framework. The purpose of this section is to propose a solution to this model-agnostic fault detection problem, by exploiting the information generated by the learning-based PI Algorithm 2.

To detect faults using this information, consider that the output of Algorithm 2 is given by

$$X_o = Q^{-1}, Z_o = Q^{-1}G. \quad (32)$$

By solely using this output, we can design the following data-based filter, for all $t \geq 0$:

$$\begin{aligned} \dot{\Sigma}(t) &= -\beta (\Sigma(t) - x^T(t)X_o x(t)) + 2x^T(t)Z_o v(t) \\ &\quad - x^T(t)(X_o V X_o + 2Z_o K)x(t), \\ \Sigma(0) &= x_0^T X_o x_0 \end{aligned} \quad (33)$$

where $\beta > 0$ and $\Sigma(t) \in \mathbb{R}$. Evidently, this filter is designed in a data-based sense, as it uses only data from the trajectories of $x(\cdot)$ and $v(\cdot)$ as well as the matrices X_o and Z_o generated by Algorithm 2. Define now the filter error

$$\tilde{\Sigma}(t) := \Sigma(t) - x^T(t)X_o x(t).$$

In the following theorem, it is shown that by checking the value of this error, one can accurately always detect actuator faults.

Theorem 6: Consider the trajectories of x given by (7), (8), where the control input $v(\cdot)$ is as in (2), and let a be (piecewise) continuous on $t \geq 0$. Consider also the filter (32) and (33). Then

- 1) If $\tilde{\Sigma} \neq 0$ then $a \neq 0$.
- 2) If $\tilde{\Sigma} \equiv 0$, then

$$x^T(t)Q^{-1}GBa(t) = 0 \quad (34)$$

(almost) everywhere on $t \geq 0$.

Remark 8: According to the condition (34) provided by Theorem 6, practically all actuator faults can be detected using the filter (33), and without knowledge of the system's matrices A and G . The only kind of fault that can go completely undetectable is if $a(t) = -u(t)$ while $x_0 = 0$, i.e., if all actuators suffer from a failure while the system is at the origin. In this scenario, $x(t) \equiv 0$ for all $t \geq 0$, hence $\tilde{\Sigma}(t) \equiv 0$ for all $t \geq 0$. However, this fault would not disrupt the goal of keeping the system regulated at the origin, and the corresponding failure could be detected by simply inspecting the values of $u(t)$ and $x(t)$, $\forall t \geq 0$. $\square$

Remark 9: Note that the proposed data-based method of detecting actuation faults enjoys similarities with the detection mechanism introduced in [29]. Specifically, the solution to an ARE, as well as the corresponding LQR gain, were also employed in [29] to detect exogenous inputs by exploiting the principle of optimality. However, the detection scheme proposed in this article significantly improves [29] in two ways; first, as seen in (33), no integrals need to be computed in a receding horizon in order to detect exogenous inputs, thus significantly reducing computational complexity; and second, with the detection filter employed here, the control input $v(\cdot)$ used in the closed loop can be arbitrary, and not necessarily the LQR control law corresponding to the ARE. $\square$

Similar to most fault detection mechanisms, if the system at hand is under the effect of input or measurement noise, the ideal results of Theorem 6 will not hold. In this case, the appropriate course of action would be to specify a small detection threshold

$\eta > 0$ depending on the severity of the input/measurement noise and declare the presence of a fault only if the detection signal passes this threshold. It should be pointed out, however, that a good choice for the value of the threshold is crucial in minimizing false positives and negatives and requires good intuition and engineering. The following discussion provides a few insights in this regard.

Consider the uncertain, noisy version of system (7)

$$\begin{aligned} \dot{x}(t) &= Ax(t) + G(v_a(t) + n_u(t)), \quad x(0) = x_0, \\ \bar{x}(t) &= x(t) + n_s(t) \end{aligned}$$

where $n_u(t) \in \mathbb{R}^m$, $n_s(t) \in \mathbb{R}^n$ denote input and measurement noise such that $\|n_u(t)\| \leq \bar{n}_u$, $\|n_s(t)\| \leq \bar{n}_s$, $t \geq 0$, $\bar{n}_u, \bar{n}_s \geq 0$. Consider also the following noisy version of the filter (33):

$$\begin{aligned} \dot{\Sigma}_\eta(t) &= -\beta (\Sigma_\eta(t) - \bar{x}^T(t)\bar{X}_o \bar{x}(t)) + 2\bar{x}^T(t)\bar{Z}_o v(t) \\ &\quad - \bar{x}^T(t)(\bar{X}_o V \bar{X}_o + 2\bar{Z}_o K)\bar{x}(t) \end{aligned} \quad (35)$$

where $\bar{X}_o \in \mathbb{R}^{n \times n}$, $\bar{Z}_o \in \mathbb{R}^{n \times m}$ are noisy estimates of X_o , Z_o such that $\|\bar{X}_o - X_o\|_F \leq \Delta X_o$, $\|\bar{Z}_o - Z_o\|_F \leq \Delta Z_o$, $\Delta X_o, \Delta Z_o \geq 0$. Defining the detection signal

$$\tilde{\Sigma}_\eta(t) = \Sigma_\eta(t) - \bar{x}^T(t)\bar{X}_o \bar{x}(t)$$

one should declare a fault as long as $|\tilde{\Sigma}_\eta| \geq \eta$ for a properly chosen threshold $\eta > 0$. To get an intuition about this threshold, let $\tilde{\Sigma}_\eta(t) = \Sigma_\eta(t) - \bar{x}^T(t)\bar{X}_o \bar{x}(t)$. Following the same analysis as in the proof of Theorem 6, we have

$$\begin{aligned} \frac{d(x^T(t)X_o x(t))}{dt} &= -x^T(t)(X_o V X_o + 2Z_o K)x(t) \\ &\quad + 2x^T(t)Z_o(v_a(t) + n_u(t)) \end{aligned}$$

and hence

$$\begin{aligned} \dot{\tilde{\Sigma}}_\eta(t) &= -\beta \tilde{\Sigma}_\eta(t) - 2x^T(t)Z_o B a(t) - 2x^T(t)Z_o n_u(t) \\ &\quad + \beta(\bar{x}^T(t)\bar{X}_o \bar{x}(t) - x^T(t)X_o x(t)) + 2(\bar{x}^T(t)\bar{Z}_o - x^T(t)Z_o)v \\ &\quad - \bar{x}^T(t)(\bar{X}_o V \bar{X}_o + 2\bar{Z}_o K)\bar{x}(t) + x^T(t)(X_o V X_o + 2Z_o K)x(t). \end{aligned} \quad (36)$$

The first two terms at the right-hand side of (36) are identical to those of the ideal error equation (56), whereas the rest of the terms are estimation-error/noise-based terms that can be upper bounded (after some tedious calculations) as functions dependent on the upper bounds of the estimation errors ΔX_o , ΔZ_o , and of the noise $\bar{n}_u$, $\bar{n}_s$. It then follows, in the absence of a fault and as long as the state and control signals are uniformly bounded, that $\tilde{\Sigma}_\eta$ is uniformly ultimately bounded, with an ultimate bound dependent on ΔX_o , ΔZ_o , $\bar{n}_u$, and $\bar{n}_s$. In addition, for the detection signal we have $\tilde{\Sigma}_\eta = \tilde{\Sigma}_\eta + x^T(t)X_o x(t) - \bar{x}^T(t)\bar{X}_o \bar{x}(t)$, where $x^T(t)X_o x(t) - \bar{x}^T(t)\bar{X}_o \bar{x}(t)$ is another error term whose upper bound directly depends on $\bar{n}_s$ and ΔX_o . Hence, in the presence of noise and estimation errors, the threshold η can be chosen by appending all the aforementioned ultimate bounds.

Eq. (36) also provides intuition regarding the choice of the parameter β of the filter. In the ideal, noise-free version of the detector, one would set β to be as large as possible, to quickly verify the presence or the absence of a fault. However, in the presence of measurement noise and estimation errors, selecting a large value for β can increase the magnitude of the error

term $\beta(\bar{x}^T(t)\bar{X}_o\bar{x}(t) - x^T(t)X_ox(t))$ in (36). In that respect, intuition and engineering should be used to select an appropriate value for β , so as to minimize the effect of noise on $\bar{\Sigma}_\eta$ while still rendering the filter quick enough in detecting the presence or absence of faults.

VII. EXTENSIONS AND LIMITATIONS

In this section, we will discuss any limitations of the methods of this article and seek directions to overcome them.

A. Extensions to Cases With a Positive Semidefinite Covariance Matrix V

In the formulation of the actuator selection metrics $f_p(\cdot)$ and $f_{pr}(\cdot)$ in Sections IV-C and IV-D, it has been assumed that the covariance matrix $V = \mathbb{E}[x_0x_0^T]$ of the initial state vector is positive-definite. In case that V is positive semidefinite only, it is straightforward to verify that the results of Theorem 1 still hold, but with the differentiation that $Q \succeq 0$ instead of $Q \succ 0$. This implies that Q is not necessarily invertible if $V \succeq 0$, hence rendering the learning-based actuator selection procedure of Section V inapplicable. This is one limitation of this article.

To deal with the case where $V \succeq 0$, the following Proposition provides a method to reexpress Q as the difference of two positive-definite matrices Q_1, Q_2 , both of which satisfy LEs similar to the LE that characterizes Q . In that respect, Q_1 and Q_2 , and subsequently Q , can be readily learnt using the methods of Section V.

Proposition 2: Assume that initial state covariance V given by (13) is positive semidefinite. Define $V_1 = V + \bar{V}$, $V_2 = \bar{V}$ for some user-specified matrix $\bar{V} \succ 0$. Then

$$f_p(B) = \text{tr}(QR) = \text{tr}(Q_1R) - \text{tr}(Q_2R)$$

where $Q \succeq 0$ uniquely solves (17), and $Q_1, Q_2 \succ 0$ uniquely solve the LEs

$$(A + GK)Q_1 + Q_1(A + GK)^T + V_1 = 0, \quad (37)$$

$$(A + GK)Q_2 + Q_2(A + GK)^T + V_2 = 0. \quad (38)$$

In addition, $Q = Q_1 - Q_2$.

Since the matrices Q_1 and Q_2 are strictly positive-definite, they can be readily learnt using the methods of Section V, hence allowing us to evaluate $f_p(B)$ even in the case where V is positive semidefinite only. However, it is not possible here to obtain a relaxation of $f_p(\cdot)$ in the form of (20), because Q is singular. A heuristic could be to add a small regulation term to the matrix Q (or V) so that it becomes positive-definite, and subsequently obtain a relaxation of $f_p(\cdot)$ in the form of (20). However, this action will lead to loss of optimality. On the other hand, approaches based on the relaxation (10) and (11) are still valid.

B. Extensions to a Class of Nonlinear Systems

The actuator selection metrics formulated in Sections IV-A and IV-B, which quantify the actuation energy expended on average, are applicable even in the case where the system's dynamics are nonlinear. However, the same is not true for the more direct metrics of Sections IV-C and IV-D, which directly quantify the energy expended in the closed loop over an infinite horizon. In this regard, we provide a few insights on how these metrics can be extended to the nonlinear case.

Assume that instead of (1), we have a nonlinear system of the form

$$\dot{x}(t) = f(x(t)) + g(x(t))v(t), \quad x(0) = x_0, \quad t \geq 0 \quad (39)$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $g: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are locally Lipschitz functions with $f(0) = 0$, and $v(t) \in \mathbb{R}^m$ denotes again the high-level control input, satisfying the mapping from actuator commands (2). In an analogous manner to the linear case, assume that the desired control policy is given by $v(t) = s(x(t))$, where $s: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a continuous state-feedback control law that asymptotically stabilizes (39) for any $x_0 \in \mathbb{R}^n$, while expending a finite amount of energy, i.e. $\int_0^\infty \|s(x(t))\|^2 dt < \infty$ over the trajectories of (39) under $v(t) = s(x(t))$. Following the optimal control allocation strategy (4), the energy (14) expended over an infinite horizon is:

$$f_p(B) = \mathbb{E} \left[\int_0^\infty s^T(x(t))(BW^{-1}B^T)^{-1}s(x(t))dt \right] \quad (40)$$

where $\mathbb{E}$ is taken with respect to the distribution of x_0 , and the integration is over the trajectories of (39) under the stabilizing control $s(\cdot)$. The main issue here is that (40) cannot be expressed in a static form using LEs, owing to the nonlinear nature of the dynamics (39). Nevertheless, Lyapunov-like partial differential equations (PDEs) can be employed in this direction, given the following regularity assumption:

Assumption 2: Each entry of the matrix function $V(x_0) := \int_0^\infty s(x(t))s^T(x(t))dt$ is continuously differentiable. $\square$

Proposition 3: Let Assumption 2 hold. For all $i, j \in \{1, \dots, m\}$, denote $V_{i,j}(\cdot) = [V(\cdot)]_{i,j}$, $s_i(\cdot) = [s(\cdot)]_i$, and $s_j(\cdot) = [s(\cdot)]_j$. Then the metric (40) can be expressed as

$$f_p(B) = \text{tr}(\mathbb{E}[V(x_0)](BW^{-1}B^T)^{-1}) \quad (41)$$

where each entry $V_{i,j}$ of V uniquely solves the PDE

$$\nabla_x V_{i,j}^T(x)(f(x) + g(x)s(x)) + s_i(x)s_j(x) = 0, \quad V_{i,j}(0) = 0. \quad (42)$$

Remark 10: Notably, optimizing $f_p(B)$ in (41) with respect to B (in the sense of (6)) can be a computationally hard task, owing to the appearance of the inverse $(BW^{-1}B^T)^{-1}$. However, following the same line of analysis as in Section IV-D, a relaxed version of (41) can be defined as

$$f_{pr}(B) = -\text{tr}(\mathbb{E}[V(x_0)]^{-1}BW^{-1}B^T) \quad (43)$$

which is optimizable in $\mathcal{O}(N \log N)$ time. $\square$

C. Limitations

The solutions $V_{i,j}$ of the PDEs (42), $i, j \in \{1, \dots, m\}$ can be solved without knowledge of the system's dynamics functions f, g , in an approximate sense, using actor-critic methods; the specific details can be found in [43] and [44]. Consequently, the actuator evaluation metrics (41), (43) can also be optimized in a model-free sense, but a couple of limitations exist with respect to the linear system case as follows.

- 1) The need to employ m^2 approximation structures to approximate $V_{i,j}$, $i, j \in \{1, \dots, m\}$. Such structures inherently work only locally, and significantly increase computational complexity.
- 2) The need to know exactly the probability distribution for x_0 , instead of just its covariance, in order to compute $\mathbb{E}[V(x_0)]$.

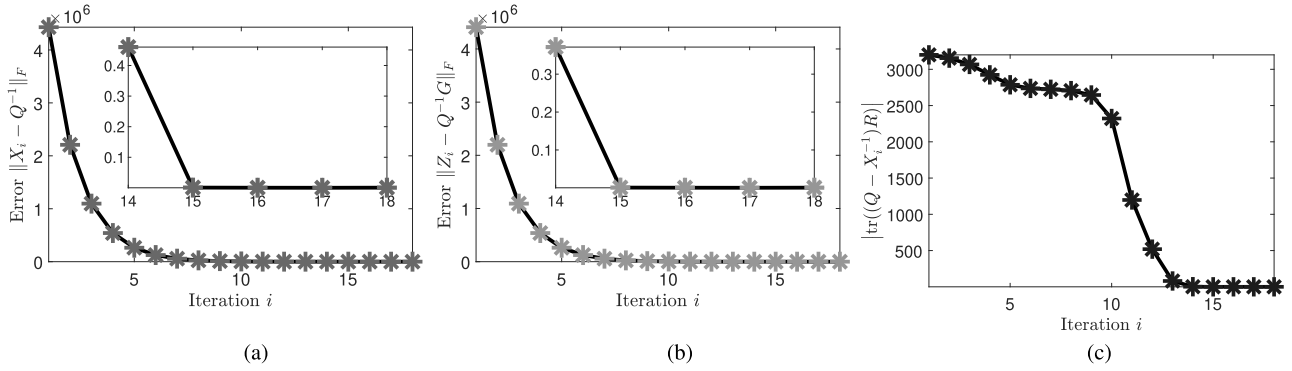


Fig. 1. Evolution of the estimation errors in each iteration of learning-based PI. (a) The error norm $\|X_i - Q^{-1}\|_F$, $i \in \mathbb{N}$. (b) The error norm $\|Z_i - Q^{-1}G\|_F$, $i \in \mathbb{N}$. (c) The error $|\text{tr}((Q - X_i^{-1})R)|$ in the approximation of the optimal actuation energy, $i \in \mathbb{N}$.

While these limitations cannot easily be dispensed with, they are natural owing to the nonlinear nature of (39).

VIII. SIMULATIONS

We consider the linearized model of an electric vertical takeoff and landing (eVTOL) aircraft [45], with $n = 9$ states and $m = 5$ high-level control inputs. Its input matrix is given by $G = \{e_4, e_5, e_6, e_7, e_9\}$, where $e_i \in \mathbb{R}^9$ is a unit vector with i th entry equal to unity, so that only states 4 – 7 and 9 are actuated. The set $\mathcal{S} = \{s_1, s_2, \dots, s_{14}\}$ of available actuators is shown at the bottom of this page, while a full description of the state matrix A can be found in [45].

The initial set of actuators is $B_1 = [s_1 \ s_2 \ s_3 \ s_{13} \ s_{14}]$, and we want to augment it with $k_2 = 6$ additional actuators from $\mathcal{S}$. To do this, we solve the optimization problem (6), where the cost function f is given by (18), the matrix K is a stabilizing gain, the covariance matrix is $V = 10^{-3}I_9$, and the weight for each actuator is $w_s = 1$, for all $s \in \mathcal{S}$. To learn the matrix Q of the cost (18) without knowledge of the matrices A and G , Algorithm 2 is applied after gathering data from the system for a period of 9 seconds using the control input $v(t) = Kx(t) + \eta(t)$, where each entry of $\eta(t)$ contains 100 sinusoids imposing exploration, each with an amplitude uniformly chosen over $[0, 0.01]$, and a frequency uniformly chosen over $[-50, 50]$. The exploration term $\eta(t)$ is then terminated at $t = 9$ s, after which point the control input simplifies to $v(t) = Kx(t)$.

The evolution of the norm of the trajectories during the data-gathering phase is shown in Fig. 2, while Fig. 1(a) and (b) show the distance of the sequences $\{X_i\}_{i \in \mathbb{N}}$, $\{Z_i\}_{i \in \mathbb{N}}$

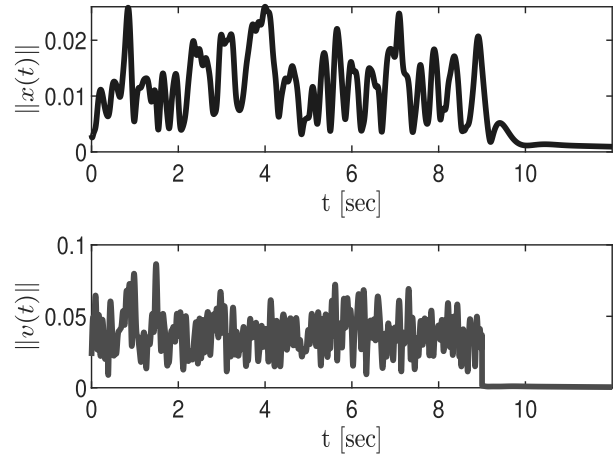


Fig. 2. Evolution of $\|x(t)\|$ and $\|v(t)\|$ during the data-gathering phase. We notice that the state and the control input were perturbed only slightly in order to gather enough data for Algorithm 2.

from the matrices Q^{-1} , $Q^{-1}G$, respectively. It can be seen that after 15 iterations, these distances practically vanish, and $X_{15} \approx Q^{-1}$, $Z_{15} \approx Q^{-1}G$. Subsequently, the matrix X_{18}^{-1} is used as a substitute of Q and the actuator placement optimization problem (6) is solved. The optimal solution is found to be $B^* = \{s_2, s_3, s_4, s_6, s_7, s_{11}\}$, which yields an optimal average actuation energy value of $\text{tr}(QR) = 4.8 \cdot 10^3$. The approximation error $|\text{tr}((Q - X_i^{-1})R)|$ of this optimal actuation energy

$$\mathcal{S} = \left\{ \begin{pmatrix} 0.6825 \\ 0.8888 \\ -0.1503 \\ -0.3510 \\ -0.6197 \end{pmatrix}, \begin{pmatrix} 10^{-4} \\ 10^{-4} \\ 0 \\ 10^{-4} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 10^{-4} \\ 0 \\ 10^{-4} \\ 0 \end{pmatrix}, \begin{pmatrix} -0.6825 \\ 0.8888 \\ 0.1503 \\ -0.3510 \\ -0.6197 \end{pmatrix}, \begin{pmatrix} -10^{-4} \\ 10^{-4} \\ 0 \\ 10^{-4} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 10^{-4} \\ 0 \\ 10^{-4} \\ 0 \end{pmatrix}, \begin{pmatrix} 1.2330 \\ -0.2469 \\ -0.0419 \\ -0.0332 \\ -0.4254 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 10^{-4} \\ 10^{-4} \\ 0 \end{pmatrix}, \right. \\ \left. \begin{pmatrix} 0 \\ 0 \\ 10^{-4} \\ 10^{-4} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 10^{-4} \\ 10^{-4} \\ 0 \end{pmatrix}, \begin{pmatrix} -1.2330 \\ -0.2469 \\ 0.0419 \\ -0.0332 \\ -0.4254 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -10^{-4} \\ 10^{-4} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -10^{-4} \\ 10^{-4} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 10^{-4} \\ 10^{-4} \\ 0 \end{pmatrix} \right\}$$

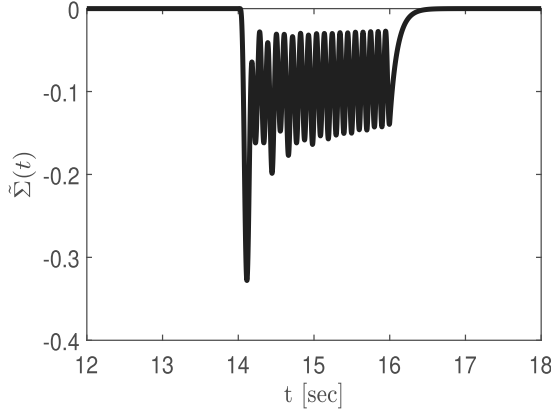


Fig. 3. Evolution of the residual $\tilde{\Sigma}(t)$ for $t \in [12, 18]$. The residual is immediately disturbed as soon as the actuator fault appears, and converges back to zero rapidly when the fault vanishes.

value per iteration of Algorithm 2 is also shown in Fig. 1(c), where it evidently converges to 0 as the number of iterations increases.

Next, we investigate the efficiency of the proposed data-based actuator selection approach in dealing with model uncertainties. In this direction, we assume that there is knowledge of the system matrices A and G , but the entries of matrix A are known only up to an error of ± 0.01 . Then, under the same choice of parameters as previously and using a model-based approach, the estimated optimal set of actuators is $\{s_3, s_4, s_5, s_6, s_7, s_{11}\}$, where we notice that the second actuator has been substituted in favor for the fifth. In addition, the estimated minimum value of the average energy (18) is 13×10^3 , which significantly differs from the true optimal value 4.8×10^3 . Therefore, we conclude that, in the quest to find the best set of actuators for the system, using input–output data and performing Algorithm 2 is superior to using even slightly inaccurate prior knowledge about the system.

Finally, we proceed to test the operation of the fault detection mechanism of Section VI. Specifically, we employ for all $t \in [12, 18]$ the filter (33) with damping rate $\beta = 10$, and with the matrices X_o and Z_o set to be equal to the output of Algorithm 2, i.e., $X_o = X_{18}$ and $Z_o = Z_{18}$. To verify the efficacy of this data-based filter, we assume that the first actuator of the system is under the effect of a fault over $t \in [14, 16]$, so that the first entry of $a(t)$ is equal to $0.1\cos(t^2)$ over $t \in [14, 16]$ and zero everywhere else. The results are shown in Fig. 3, where it can be seen that the value of the residual $\tilde{\Sigma}$ becomes nonzero as soon as the fault begins affecting the first actuator. In addition, when the effect of the fault disappears, $\tilde{\Sigma}$ converges rapidly back to zero.

IX. CONCLUSION

We considered the problem of data-based actuator selection in the context of optimal CA. Specifically, two metrics for evaluating the efficiency of actuators were proposed, based on the value of the minimum energy required to achieve a given control objective in the CA problem. Subsequently, a data-driven method was proposed to perform the corresponding actuator selection procedure model-free, without knowledge of the system's state and input matrices. This task was achieved by successively approximating the antistabilizing solution to an associated ARE using trajectory data. Finally, the solution to

this ARE was used to perform data-based fault detection, with complexity lower than that of similar methods in the literature.

Future work will include an extension to a completely adaptive system, where both learning, actuator selection, and control allocation take place in real time.

APPENDIX

A. Proof of Proposition 1

One possible weighted controllability Gramian for the control pair (E, B) is given by

$$\int_0^1 e^{E\tau} B W^{-1} B^T e^{E^T \tau} d\tau = \int_0^1 B W^{-1} B^T d\tau = B W^{-1} B^T$$

where we used the fact that $e^{E\tau} = I_m$ for any $\tau \geq 0$. ■

B. Proof of Theorem 1

The system (1) in closed loop with the control policy (12) is $\dot{x}(t) = (A + GK)x(t)$. Then, we have

$$\begin{aligned} f_p(B) &= \mathbb{E} \left[\int_0^\infty x^T(t) K^T (B W^{-1} B^T)^{-1} K x(t) dt \right] \\ &= \mathbb{E} \left[\int_0^\infty x_0^T e^{(A+GK)^T t} R e^{(A+GK)t} x_0 dt \right] \\ &= \mathbb{E} \left[\text{tr} \int_0^\infty e^{(A+GK)^T t} R e^{(A+GK)t} dt x_0 x_0^T \right] \\ &= \text{tr} \int_0^\infty e^{(A+GK)^T t} R e^{(A+GK)t} dt \mathbb{E} [x_0 x_0^T] \\ &= \text{tr}(PV) \end{aligned}$$

where $P = \int_0^\infty e^{(A+GK)^T t} R e^{(A+GK)t} dt$ satisfies the LE (16) because $A + GK$ is Hurwitz [46]. Since $R \succeq 0$, one also has $P \succeq 0$. Similarly

$$\begin{aligned} f_p(B) &= \mathbb{E} \left[\int_0^\infty x^T(t) K^T (B W^{-1} B^T)^{-1} K x(t) dt \right] \\ &= \mathbb{E} \left[\int_0^\infty x_0^T e^{(A+GK)^T t} R e^{(A+GK)t} x_0 dt \right] \\ &= \mathbb{E} \left[\text{tr} \int_0^\infty e^{(A+GK)t} x_0 x_0^T e^{(A+GK)^T t} dt R \right] \\ &= \text{tr} \int_0^\infty e^{(A+GK)t} \mathbb{E} [x_0 x_0^T] e^{(A+GK)^T t} dt R \\ &= \text{tr}(QR) \end{aligned}$$

where $Q = \int_0^\infty e^{(A+GK)t} V e^{(A+GK)^T t} dt$ satisfies the LE (17) because $A + GK$ is Hurwitz [46]. Note that Q is positive-definite because V is also positive-definite. ■

C. Proof of Lemma 1

Both items will be verified by inspecting the stability properties of the matrix $\tilde{A}(X) = (A + GK) + VX$, per Definition 2.

For the first item, it can be seen by inspection that $X_s = 0$ is a solution to (21). Additionally, one has $\tilde{A}(0) = A + GK$, which is a Hurwitz matrix by assumption, hence $X_s = 0$ is indeed the stabilizing solution to (21).

For the second item, it can be seen that $X_a = Q^{-1}$ is a solution to (21) by pre- and postmultiplying (17) by Q^{-1} . Moreover, a negation and a further manipulation of (21) yields

$$-(A + GK + VX)^T X - X(A + GK + VX) + XVX = 0. \quad (44)$$

Notice that (44) can be seen as an LE for the matrix $-(A + GK + VX)$. Additionally, for $X = X_a = Q^{-1}$, both the matrices X and XVX in the LE are strictly positive-definite, i.e., $-(A + GK + VQ^{-1})$ is Hurwitz [46]. This in turn implies that $\tilde{A}(Q^{-1}) = (A + GK) + VQ^{-1}$ is a matrix whose eigenvalues have strictly positive real parts. ■

D. Proof of Lemma 2

Q^{-1} is a solution to (23) due to the fact that (23) is just the negation of (21). In addition, $-(A + GK) - VQ^{-1}$ is Hurwitz owing to Lemma 1, hence Q^{-1} is stabilizing with respect to $-(A + GK)$. ■

E. Proof of Theorem 2

Parts of the proof follow from [39] and [40], and from the fact that Q^{-1} is the stabilizing solution to (23) due to Lemma 2. However, since no constant term is involved in the ARE (23), 1) and 3) need to be proved. We will use induction for this purpose.

For $i = 0$, the matrix $-(A + GK + Y_0)$ is Hurwitz by construction. For $i \in \mathbb{N}$, let us assume that $-(A + GK + Y_i)$ is Hurwitz. Then, since (24) is an LE for $-(A + GK + Y_i)$ and $Y_i^T V^{-1} Y_i$ is positive semidefinite, it follows that X_i will be strictly positive-definite, if the pair $(-(A + GK + Y_i), V^{-\frac{1}{2}} Y_i)$ is observable [46]. If we suppose that this pair is not observable, then from [46] there exists $\lambda \in \mathbb{C}$ such that

$$\text{rank} \begin{bmatrix} \lambda I_n + (A + GK + Y_i) \\ V^{-\frac{1}{2}} Y_i \end{bmatrix} < n. \quad (45)$$

Therefore we can find a nonzero vector $v \in \mathbb{C}^n$ such that

$$(\lambda I_n + (A + GK + Y_i))v = 0, \quad V^{-\frac{1}{2}} Y_i v = 0$$

from which we derive the equivalent set of equations

$$-(A + GK + Y_i)v = \lambda v, \quad -(A + GK)v = \lambda v. \quad (46)$$

Eq. (46) imply that λ is an eigenvalue of both $-(A + GK + Y_i)$ and $-(A + GK)$. However, $-(A + GK + Y_i)$ is Hurwitz, hence its eigenvalues have strictly negative real parts. On the other hand, $(A + GK)$ is also Hurwitz, hence the eigenvalues of $-(A + GK)$ have strictly positive real parts. We therefore conclude that there cannot exist $\lambda \in \mathbb{C}$ such that (45) holds, hence $(-(A + GK + Y_i), V^{-\frac{1}{2}} Y_i)$ is indeed observable and X_i is strictly positive-definite. Subsequently, a manipulation of (24) yields

$$-(A + GK + Y_{i+1})^T X_i - X_i(A + GK + Y_{i+1}) + (V^{\frac{1}{2}} X_i - V^{-\frac{1}{2}} Y_i)^T (V^{\frac{1}{2}} X_i - V^{-\frac{1}{2}} Y_i) + X_i V X_i = 0.$$

Since X_i was proved to be strictly positive-definite, the above LE implies that $-(A + GK + Y_{i+1})$ is Hurwitz, which concludes the induction and proves item 1).

For 2), given that each matrix in the sequence $\{-(A + GK + Y_i)\}_{i \in \mathbb{N}}$ has been proved to be Hurwitz, the proof follows from [40].

For 3), notice that $X_\infty = \lim_{i \rightarrow \infty} X_i$ exists because $\{X_i\}_{i \in \mathbb{N}}$ is a decreasing sequence bounded below, and X_∞ satisfies the ARE (23). Therefore, (23) for $X = X_\infty$ yields

$$-(A + GK + VQ^{-1})^T X_\infty - X_\infty(A + GK + VQ^{-1}) - X_\infty V X_\infty + X_\infty V Q^{-1} + Q^{-1} V X_\infty = 0. \quad (47)$$

In addition, Q^{-1} is another solution to (23), hence

$$-(A + GK + VQ^{-1})^T Q^{-1} - Q^{-1}(A + GK + VQ^{-1}) + Q^{-1} V Q^{-1} = 0. \quad (48)$$

Subtracting (47) from (48) yields

$$-(A + GK + VQ^{-1})^T (Q^{-1} - X_\infty) - (Q^{-1} - X_\infty)(A + GK + VQ^{-1}) + (X_\infty - Q^{-1})V(X_\infty - Q^{-1}) = 0. \quad (49)$$

Note that $(X_\infty - Q^{-1})V(X_\infty - Q^{-1}) \geq 0$. In addition, (49) is an LE for the matrix $-(A + GK + VQ^{-1})$, which is Hurwitz because Q^{-1} is the stabilizing solution to (23) with respect to $-(A + GK)$. Hence, it follows that $(Q^{-1} - X_\infty) \geq 0$, or equivalently $X_\infty \preceq Q^{-1}$. On the other hand, from 2) of the Theorem we have $X_\infty \succeq Q^{-1}$, hence $X_\infty \succeq Q^{-1} \succeq X_\infty$. This means that $\lim_{i \rightarrow \infty} X_i = Q^{-1}$. ■

F. Proof of Theorem 3

Let $v \in \mathbb{C}^n$ be an eigenvector of $A + GK$, and $\lambda \in \mathbb{C}$ the corresponding eigenvalue. Then:

$$(-(A + GK) - Y_0)v = -(A + GK)v + \alpha v = (\alpha - \lambda)v.$$

Hence, $\alpha - \lambda$ is an eigenvalue of $-(A + GK) - Y_0$. Since $\text{Re}(\alpha - \lambda) \leq \alpha - \alpha^* < 0$, it follows that $-(A + GK) - Y_0$ is Hurwitz. ■

G. Proof of Lemma 3

Let $\bar{X} \in \mathbb{R}^{n \times n}$, $\bar{Z} \in \mathbb{R}^{n \times m}$, with $\bar{X}$ symmetric, satisfy

$$\Theta_i \begin{bmatrix} \text{vecs}(\bar{X}) \\ \text{vec}(\bar{Z}) \end{bmatrix} = 0. \quad (50)$$

Dropping the dependence of $x(t)$ and $v(t)$ on t , we have

$$-\frac{d}{dt}(x^T \bar{X} x) = -x^T (A^T \bar{X} + \bar{X} A)x - 2x^T \bar{X} G v.$$

After adding and subtracting identical terms, this turns into

$$\begin{aligned} & -\frac{d}{dt}(x^T \bar{X} x) - x^T \bar{X} Y_i x - x^T Y_i^T \bar{X} x + 2x^T \bar{Z} v - 2x^T \bar{Z} K x \\ & = -x^T ((A + GK + Y_i)^T \bar{X} + \bar{X} (A + GK + Y_i) \\ & \quad + (\bar{Z} - \bar{X} G)K + K^T (\bar{Z} - \bar{X} G)^T)x + 2x^T (\bar{Z} - \bar{X} G)v. \end{aligned}$$

Integrating this equation over $[t_\kappa, t_\kappa + T]$, $\kappa \in \{0, \dots, K_0\}$, and stacking it, we obtain

$$\Theta_i \begin{bmatrix} \text{vecs}(\bar{X}) \\ \text{vec}(\bar{Z}) \end{bmatrix} = \begin{bmatrix} J_{xx} & J_{vx} \end{bmatrix} \begin{bmatrix} \text{vec}(M_1) \\ \text{vec}(M_2) \end{bmatrix} \quad (51)$$

where

$$\begin{aligned} M_1 &= -(A + GK + Y_i)^T \bar{X} - \bar{X}(A + GK + Y_i) \\ &\quad - (\bar{Z} - \bar{X}G)K - K^T(\bar{Z} - \bar{X}G)^T, \\ M_2 &= 2(\bar{Z} - \bar{X}G). \end{aligned}$$

Using the fact that M_1 is symmetric, (51) can be rewritten as

$$\Theta_i \begin{bmatrix} \text{vecs}(\bar{X}) \\ \text{vec}(\bar{Z}) \end{bmatrix} = \begin{bmatrix} \bar{J}_{xx} & J_{vx} \end{bmatrix} \begin{bmatrix} \text{vecs}(M_1) \\ \text{vec}(M_2) \end{bmatrix} \quad (52)$$

where $\bar{J}_{xx} = [J_0 \dots J_{K_0}]^T$ and $J_\kappa = \int_{t_\kappa}^{t_\kappa + T} \text{vech}(\text{vec}^{-1}(x \otimes x))dt$, $\kappa \in \{0, \dots, K_0\}$. Note here that if (30) is true, we must also have $\text{rank}[\bar{J}_{xx} \ J_{vx}] = \frac{n(n+1)}{2} + nm$. Therefore, from (50) and (52) we conclude that $M_1 = 0$, $M_2 = 0$. From $M_2 = 0$ we conclude that $\bar{Z} = \bar{X}G$, hence from $M_1 = 0$ we also conclude that $-(A + GK + Y_i)^T \bar{X} - \bar{X}(A + GK + Y_i) = 0$. But by Theorem 2, $-(A + GK + Y_i)$ is Hurwitz hence this LE has $\bar{X} = 0$ as a unique solution. This also implies that $\bar{Z} = 0$ by the fact that $\bar{Z} = \bar{X}G$. Therefore, given condition (30), the only solution to (50) is the trivial one, hence the kernel of Θ_i is trivial for all $i \in \mathbb{N}$. ■

H. Proof of Theorem 4

Given condition (30) and from Lemma 3, (29) has a unique solution that is given by (31). Additionally, any matrix X_i satisfying (24) also satisfies (29) together with $Z_i = X_i G$, for all $i \in \mathbb{N}$. Hence, the proof follows directly from Theorem 2 and (26). ■

I. Proof of Theorem 5

Consider the nominal sequence $\{X_i\}_{i \in \mathbb{N}}$ of Procedure 1 under $Y_0 = \hat{Y}_0$. Then, we have that $[\mathcal{G}(X_0)]_{uu} = V^{-1} \succ 0$ is invertible, and has strictly positive eigenvalues. For the perturbed sequence, since $Y_0 = \hat{Y}_0$ we have $\hat{X}_0 = X_0$, and thus $\hat{\mathcal{G}}_0 = \mathcal{G}(X_0) + \Delta\mathcal{G}_0$. By the continuity of the eigenvalues, we can find $\bar{\beta}_1 > 0$, such that if $\|\Delta\mathcal{G}\|_\infty < \bar{\beta}_1$ then $[\hat{\mathcal{G}}_0]_{uu}$ is invertible. Given that this condition holds, we have by [47]

$$\begin{aligned} \hat{Y}_1 &= [\hat{\mathcal{G}}_0]_{uu}^{-1}[\hat{\mathcal{G}}_0]_{ux} = [\mathcal{G}(X_0) + \Delta\mathcal{G}_0]_{uu}^{-1}[\mathcal{G}(X_0) + \Delta\mathcal{G}_0]_{ux} \\ &= Y_1 - V[\Delta\mathcal{G}_0]_{uu}[\hat{\mathcal{G}}_0]_{uu}^{-1}[\mathcal{G}(X_0)]_{ux} + [\hat{\mathcal{G}}_0]_{uu}^{-1}[\Delta\mathcal{G}_0]_{ux} \end{aligned}$$

which implies $\hat{Y}_1 = Y_1 + \Delta Y_1(\Delta\mathcal{G})$ where $\|\Delta Y_1(\Delta\mathcal{G})\|_F \rightarrow 0$ as $\|\Delta\mathcal{G}\|_\infty \rightarrow 0$. Therefore, there exists $\hat{\beta}_1 > 0$, such that if $\|\Delta\mathcal{G}\|_\infty < \hat{\beta}_1$ then $-(A + GK + \hat{Y}_1) = -(A + GK + Y_1) - \Delta Y_1(\Delta\mathcal{G})$ is Hurwitz.

Continuing to the next step, $\hat{X}_1$ is derived as the solution to the equation $\mathcal{H}(\mathcal{G}(\hat{X}_1), \hat{Y}_1) = 0$, or equivalently of

$$-\mathcal{A}(\hat{Y}_1)^T \hat{X}_1 - \hat{X}_1 \mathcal{A}(\hat{Y}_1) + \hat{Y}_1^T V^{-1} \hat{Y}_1 = 0$$

where $\mathcal{A}(Y) := A + GK + Y$. Since $-\mathcal{A}(\hat{Y}_1)$ is Hurwitz, the unique solution to this equation is [46]

$$\begin{aligned} \text{vec}(\hat{X}_1) &= (\mathcal{A}(\hat{Y}_1) \oplus \mathcal{A}(\hat{Y}_1))^{-T} \text{vec}(Y_1^T V^{-1} Y_1 + \Delta Y_1^T V^{-1} Y_1 \\ &\quad + Y_1^T V^{-1} \Delta Y_1 + \Delta Y_1^T V^{-1} \Delta Y_1) \\ &= (\mathcal{A}(Y_1) \oplus \mathcal{A}(Y_1) + \Delta Y_1 \oplus \Delta Y_1)^{-T} \text{vec}(Y_1^T V^{-1} Y_1 \\ &\quad + \Delta Y_1^T V^{-1} Y_1 + Y_1^T V^{-1} \Delta Y_1 + \Delta Y_1^T V^{-1} \Delta Y_1). \end{aligned}$$

But $\mathcal{A}(Y_1) \oplus \mathcal{A}(Y_1)$ is invertible since $-\mathcal{A}(Y_1)$ is Hurwitz. Therefore, from [47] we have

$$\begin{aligned} \text{vec}(\hat{X}_1) &= \text{vec}(X_1) - (\mathcal{A}(Y_1) \oplus \mathcal{A}(Y_1))^{-T} (\Delta Y_1 \oplus \Delta Y_1)^T \\ &\quad \cdot (\mathcal{A}(\hat{Y}_1) \oplus \mathcal{A}(\hat{Y}_1))^{-T} \text{vec}(Y_1^T V^{-1} Y_1) + (\mathcal{A}(\hat{Y}_1) \oplus \mathcal{A}(\hat{Y}_1))^{-T} \\ &\quad \cdot \text{vec}(\Delta Y_1^T V^{-1} Y_1 + Y_1^T V^{-1} \Delta Y_1 + \Delta Y_1^T V^{-1} \Delta Y_1) \end{aligned}$$

which implies $\hat{X}_1 = X_1 + E_1(\Delta\mathcal{G})$, where $\|E_1(\Delta\mathcal{G})\|_F \rightarrow 0$ as $\|\Delta\mathcal{G}\|_\infty \rightarrow 0$. Thus, we have $\hat{\mathcal{G}}_1 = \mathcal{G}(X_1) + F_1(\Delta\mathcal{G})$, where $F_1(\Delta\mathcal{G}) = \mathcal{G}(E_1(\Delta\mathcal{G})) + \Delta\mathcal{G}_1$. Therefore, since $[\mathcal{G}(X_1)]_{uu} = V^{-1} \succ 0$ is invertible and has strictly positive eigenvalues, there exists $\bar{\beta}_2 > 0$ such that if $\|\Delta\mathcal{G}\|_\infty < \bar{\beta}_2$ then $[\hat{\mathcal{G}}_1]_{uu}$ is also invertible. Given that this condition holds, we have

$$\begin{aligned} \hat{Y}_2 &= [\hat{\mathcal{G}}_1]_{uu}^{-1}[\hat{\mathcal{G}}_1]_{ux} = [\mathcal{G}(X_1) + F_1(\Delta\mathcal{G})]_{uu}^{-1}[\mathcal{G}(X_1) + F_1(\Delta\mathcal{G})]_{ux} \\ &= Y_2 - V[F_1(\Delta\mathcal{G})]_{uu}[\hat{\mathcal{G}}_1]_{uu}^{-1}[\mathcal{G}(X_1)]_{ux} + [\hat{\mathcal{G}}_1]_{uu}^{-1}[F_1(\Delta\mathcal{G})]_{ux} \end{aligned}$$

which implies $\hat{Y}_2 = Y_2 + \Delta Y_2(\Delta\mathcal{G})$ where $\|\Delta Y_2(\Delta\mathcal{G})\|_F \rightarrow 0$ as $\|\Delta\mathcal{G}\|_\infty \rightarrow 0$. Therefore, there exists $\hat{\beta}_2 > 0$, such that if $\|\Delta\mathcal{G}\|_\infty < \hat{\beta}_2$ then $-(A + GK + \hat{Y}_2) = -(A + GK + Y_2) - \Delta Y_2(\Delta\mathcal{G})$ is Hurwitz.

Continuing the process indefinitely and combining all derived bounds on $\|\Delta\mathcal{G}\|_\infty$, for any $i^* \in \mathbb{N}$ and $\epsilon > 0$, we can find $\gamma_1 = \gamma_1(i^*, \epsilon) > 0$ such that if $\|\Delta\mathcal{G}\|_\infty < \gamma_1$ then $[\hat{\mathcal{G}}_i]_{uu}$ is invertible, $-(A + GK + \hat{Y}_i)$ is Hurwitz, $\|\hat{X}_i - X_i\|_F < \frac{\delta_0}{2}$ and $\|\hat{Y}_i - Y_i\|_F < \frac{\delta_0}{2}$, for all $i = 0, 1, 2, \dots, i^*$. Let i^* be such that $\|X_{i^*} - Q^{-1}\|_F < \frac{\delta_0}{2}$, then it follows by the triangle inequality that $\|\hat{X}_{i^*} - Q^{-1}\|_F < \delta_0$. Since $\hat{X}_{i^*} \in \mathcal{B}_{\delta_0}(Q^{-1})$, all of the results then follow by applying Lemma 4. ■

J. Proof of Theorem 6

Note that owing to (21), one has

$$(A + GK)^T Q^{-1} + Q^{-1}(A + GK) + Q^{-1}VQ^{-1} = 0.$$

Hence, pre-, and postmultiplying this equation by the trajectories of x , and using (32), we derive

$$\begin{aligned} &x^T(t)(A^T X_o + X_o A)x(t) \\ &\quad + x^T(t)(X_o V X_o + 2Z_o K)x(t) = 0. \end{aligned} \quad (53)$$

Combining (53) with (33) yields $\forall t \geq 0$

$$\begin{aligned} \dot{\Sigma}(t) &= -\beta(\Sigma(t) - x^T(t)X_o x(t)) + 2x^T(t)Z_o v(t) \\ &\quad + x^T(t)(A^T X_o + X_o A)x(t). \end{aligned} \quad (54)$$

However, owing to (7) and (8) one has $\forall t \geq 0$

$$\begin{aligned} \frac{d(x^T(t)X_o x(t))}{dt} &= x(t)^T(A^T X_o + X_o A)x(t) \\ &+ 2x^T(t)Z_o v_a(t). \end{aligned} \quad (55)$$

Hence, combining (54) and (55), and taking (2) and (8) into account, we obtain $\forall t \geq 0$

$$\dot{\tilde{\Sigma}}(t) = -\beta \tilde{\Sigma}(t) - 2x^T(t)Z_o B a(t) \quad (56)$$

where, owing to the initialization in (33)

$$\tilde{\Sigma}(0) = 0. \quad (57)$$

Items 1 and 2 may now be proved as follows.

Item 1: If $a(t) \equiv 0$, then from (56) and (57) it is straightforward to verify that $\Sigma(t) \equiv 0$. Thus, by the contrapositive argument, if $\Sigma(t) \neq 0$ then $a(t) \neq 0$.

Item 2: An integration of (56) and (57) yields, for all $t \geq 0$

$$\tilde{\Sigma}(t) = -2 \int_0^t e^{-\beta(t-\tau)} x^T(\tau) Z_o B a(\tau) d\tau.$$

Thus, if $\tilde{\Sigma}(t) \equiv 0$, it follows from the (piece-wise) continuity of a that $x^T(t)Z_o B a(t) = 0$ (almost) everywhere. ■

K. Proof of Proposition 2

Following the same analysis as in the proof of Theorem 1, it follows that $f_p(\mathcal{B}) = \text{tr}(QR)$, where $Q \succeq 0$ uniquely satisfies the LE (17). Next, note that V_1 and V_2 are both positive-definite matrices, and $A + GK$ is Hurwitz. Hence, the LEs (37)–(38) admit unique positive-definite solutions $Q_1, Q_2 \succ 0$, respectively.

Next, subtracting (38) from (37), we obtain

$$(A + GK)(Q_1 - Q_2) + (Q_1 - Q_2)(A + GK)^T + V = 0.$$

However, this LE with respect to the matrix $Q_1 - Q_2$ is exactly the same as the LE (17), which is uniquely solved by Q . Therefore, $Q = Q_1 - Q_2$, and it follows that $f_p(\mathcal{B}) = \text{tr}(Q_1 R) - \text{tr}(Q_2 R)$. ■

L. Proof of Proposition 3

We have

$$\begin{aligned} f_p(\mathcal{B}) &= \mathbb{E} \left[\int_0^\infty s^T(x(t))(BW^{-1}B^T)^{-1}s(x(t))dt \right] \\ &= \mathbb{E} \left[\text{tr} \left(\int_0^\infty s(x(t))s^T(x(t))dt (BW^{-1}B^T)^{-1} \right) \right] \\ &= \text{tr} \left(\mathbb{E} \left[\int_0^\infty s(x(t))s^T(x(t))dt \right] (BW^{-1}B^T)^{-1} \right) \\ &= \text{tr} (\mathbb{E}[V(x_0)](BW^{-1}B^T)^{-1}) \end{aligned}$$

where $V(x_0) = \int_0^\infty s(x(t))s^T(x(t))dt$. Therefore, it follows that $V_{i,j}(x_0) = \int_0^\infty s_i(x(t))s_j(x(t))dt$, $\forall x_0 \in \mathbb{R}^n$, and using standard arguments [39], we conclude that $V_{i,j}$ is the unique solution of the PDE (42). ■

REFERENCES

- [1] O. Härkegård and S. T. Glad, "Resolving actuator redundancy—optimal control vs. control allocation," *Automatica*, vol. 41, no. 1, pp. 137–144, 2005.
- [2] P. Kolaric, V. G. Lopez, and F. L. Lewis, "Optimal dynamic control allocation with guaranteed constraints and online reinforcement learning," *Automatica*, vol. 122, 2020, Art. no. 109265.
- [3] J. Jiang and X. Yu, "Fault-tolerant control systems: A comparative study between active and passive approaches," *Annu. Rev. Control*, vol. 36, no. 1, pp. 60–72, 2012.
- [4] S. R. Wells, *Application of Sliding Mode Methods to the Design of Reconfigurable Flight Control Systems*. Davis, CA, USA: University of California, 2002.
- [5] K. A. Bordignon, *Constrained Control Allocation for Systems With Redundant Control Effectors*. Blacksburg, VA, USA: Virginia Polytechnic Institute and State University, 1996.
- [6] T. A. Johansen and T. I. Fossen, "Control allocation—a survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.
- [7] M. W. Oppenheimer, D. B. Doman, and M. A. Bolender, "Control allocation for over-actuated systems," in *Proc. IEEE 14th Mediterranean Conf. Control Automat.* 2006, pp. 1–6.
- [8] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [9] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken, NJ, USA: Wiley, 2012.
- [10] L. Zaccarian, "Dynamic allocation for input redundant control systems," *Automatica*, vol. 45, no. 6, pp. 1431–1438, 2009.
- [11] T. A. Johansen, T. I. Fossen, and P. Tøndel, "Efficient optimal constrained control allocation via multiparametric programming," *J. Guid. Control, Dyn.*, vol. 28, no. 3, pp. 506–515, 2005.
- [12] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 40–52, Mar. 2014.
- [13] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 1, pp. 91–101, Mar. 2016.
- [14] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 1, pp. 67–78, Mar. 2016.
- [15] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1686–1701, Apr. 2021.
- [16] J. Milošević, A. Teixeira, K. H. Johansson, and H. Sandberg, "Actuator security indices based on perfect undetectability: Computation, robustness, and sensor placement," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3816–3831, Sep. 2020.
- [17] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Robust and adaptive sequential submodular optimization," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 89–104, Jan. 2022.
- [18] M. Pirani, E. Nekouei, H. Sandberg, and K. H. Johansson, "A game-theoretic framework for security-aware sensor placement problem in networked control systems," in *Proc. IEEE Amer. Control Conf.*, 2019, pp. 114–119.
- [19] T. Summers, "Actuator placement in networks using optimal control performance metrics," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 2703–2708.
- [20] N. Darivandi, K. Morris, and A. Khajepour, "An algorithm for LQ optimal actuator location," *Smart Mater. Structures*, vol. 22, no. 3, 2013, Art. no. 035001.
- [21] M.-A. Belabbas, "Geometric methods for optimal sensor design," *Proc. Roy. Soc. A, Mathematical, Phys. Eng. Sci.*, vol. 472, no. 2185, 2016, Art. no. 20150312.
- [22] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*. Chelmsford, MA, USA: Courier Corporation, 2011.
- [23] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Chelmsford, MA, USA: Courier Corporation, 2012.
- [24] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Third Quarter 2009.
- [25] Y. Wang, H. Silva-Saravia, and H. Pulgar-Painemal, "Actuator placement for enhanced grid dynamic performance: A machine learning approach," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 3119–3128, Jul. 2019.
- [26] J. Du, X. Yue, J. H. Hunt, and J. Shi, "Optimal placement of actuators via sparse learning for composite fuselage shape control," *J. Manuf. Sci. Eng.*, vol. 141, no. 10, pp. 101004-1–101004-12, 2019.

- [27] F. Fotiadis and K. G. Vamvoudakis, "Learning-based actuator placement for uncertain systems," in *Proc. 60th IEEE Conf. Decis. Control*, 2021, pp. 90–95.
- [28] B. Pang, T. Bian, and Z.-P. Jiang, "Robust policy iteration for continuous-time linear quadratic regulation," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 504–511, Jan. 2022.
- [29] A. Kanellopoulos and K. G. Vamvoudakis, "A moving target defense control framework for cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 1029–1043, Mar. 2020.
- [30] F. Fotiadis, K. G. Vamvoudakis, and Z.-P. Jiang, "Data-based actuator selection for optimal control allocation," in *Proc. IEEE 61st Conf. Decis. Control*, 2022, pp. 4674–4679.
- [31] T. Summers and M. Kamgarpour, "Performance guarantees for greedy maximization of non-submodular controllability metrics," in *Proc. 18th Eur. Control Conf.*, 2019, pp. 2796–2801.
- [32] A. Olshevsky, "On a relaxation of time-varying actuator placement," *IEEE Contr. Syst. Lett.*, vol. 4, no. 3, pp. 656–661, Jul. 2020.
- [33] T. H. Summers and J. Lygeros, "Optimal sensor and actuator placement in complex dynamical networks," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 3784–3789, 2014.
- [34] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.
- [35] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Syst. Control Lett.*, vol. 100, pp. 14–20, 2017.
- [36] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*. Philadelphia, PA, USA: SIAM, 2006.
- [37] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.
- [38] G. N. Saridis and C.-S. G. Lee, "An approximation theory of optimal control for trainable manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 3, pp. 152–159, Mar. 1979.
- [39] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.
- [40] D. Kleinman, "On an iterative technique for riccati equation computations," *IEEE Trans. Autom. Control*, vol. 13, no. 1, pp. 114–115, Feb. 1968.
- [41] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 909–924, Mar. 2020.
- [42] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1702–1717, Apr. 2021.
- [43] Y. Jiang and Z.-P. Jiang, "Global adaptive dynamic programming for continuous-time nonlinear systems," *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 2917–2929, Nov. 2015.
- [44] Y. Jiang and Z.-P. Jiang, "Robust adaptive dynamic programming and feedback stabilization of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 882–893, May 2014.
- [45] E. C. Suiçmez, "Full envelope nonlinear controller design for a novel electric VTOL (eVTOL) air-taxi via INDI approach combined with CA," in Dissertation, Middle East Technical University, 2021.
- [46] D. S. Bernstein, *Matrix Mathematics*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
- [47] H. V. Henderson and S. R. Searle, "On deriving the inverse of a sum of matrices," *SIAM Rev.*, vol. 23, no. 1, pp. 53–60, 1981.



Filippos Fotiadis (Graduate Student Member, IEEE) was born in Thessaloniki, Greece. He received the Diploma (joint B.Sc. and M.Sc. degree) in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2018, and the M.Sc. degrees in aerospace engineering and mathematics, in 2022 and 2023, respectively, from the Georgia Institute of Technology, Atlanta, GA, USA, where he is currently working toward the Ph.D. degree in aerospace engineering.

His research interests include optimal and learning-based control, game theory, and their applications to cyber-physical security, as well as prescribed performance control.



Kyriakos G. Vamvoudakis (Senior Member, IEEE) was born in Athens, Greece. He received the Diploma in electronic and computer engineering from the Technical University of Crete, Crete, Greece, in 2006, and the M.Sc. and Ph.D. degrees in electrical engineering from The University of Texas, Arlington, TX, USA, in 2008 and 2011, respectively.

From 2012 to 2016, he was a Project Research Scientist with the Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA, USA. He was an Assistant Professor with the Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA, USA, until 2018. He is currently the Dutton-Ducoffe Endowed Professor with The Daniel Guggenheim School of Aerospace Engineering, Georgia Tech, Atlanta, GA, USA. He holds a secondary appointment with the School of Electrical and Computer Engineering. His research interests include reinforcement learning, control theory, cyber-physical security, bounded rationality, and safe/assured autonomy.

Dr. Vamvoudakis is the recipient of a 2018 NSF CAREER award, a 2018 DoD Minerva Research Initiative Award, a 2019 ARO YIP award, a 2021 GT Chapter Sigma Xi Young Faculty Award, and his work has been recognized with best paper nominations and several international awards including the 2016 International Neural Network Society Young Investigator Award, the Best Paper Award for Autonomous/Unmanned Vehicles at the 27th Army Science Conference in 2010, and the Best Researcher Award from the Automation and Robotics Research Institute in 2011. He is currently a Member of the IEEE Control Systems Society Conference Editorial Board, and an Associate Editor of *Automatica*; *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*; *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS*; *IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE*; *Neurocomputing*; *Journal of Optimization Theory and Applications*; *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*.



Zhong-Ping Jiang (Fellow, IEEE) received the M.Sc. degree in statistics from Paris-Saclay University, Bures-sur-Yvette, France, in 1989, and the Ph.D. degree in automatic control and mathematics from ParisTech-Mines (formerly called the Ecole des Mines de Paris), Paris, France, in 1993, under the direction of Prof. Laurent Praly.

He is currently a Professor of electrical and computer engineering with the Tandon School of Engineering, New York University, NY, USA.

He has authored or coauthored more than 500 peer-reviewed journal and conference papers, and has written six books. His main research interests include stability theory, robust/adaptive/distributed nonlinear control, robust adaptive dynamic programming, reinforcement learning, and their applications to information, mechanical, and biological systems.

Dr. Jiang is a recipient of the prestigious Queen Elizabeth II Fellowship Award from the Australian Research Council, CAREER Award from the U.S. National Science Foundation, JSPS Invitation Fellowship from the Japan Society for the Promotion of Science, Distinguished Overseas Chinese Scholar Award from the NSF of China, several best paper awards, and the Excellence in Research Award from the NYU Tandon School of Engineering in 2022. He has served as Deputy Editor-in-Chief, Senior Editor, and Associate Editor for numerous journals. He is a Fellow of the IFAC, CAA, and AAIA, a foreign member of the Academia Europaea (Academy of Europe), and is among the Clarivate Analytics Highly Cited Researchers.