On the Synthesis of Bellman Inequalities for Data-Driven Optimal Control

Andrea Martinelli, Matilde Gargiani and John Lygeros

Abstract— In the context of the linear programming (LP) approach to data-driven control, one assumes that the dynamical system is unknown but can be observed indirectly through data on its evolution. Both theoretical and empirical evidence suggest that a desired suboptimality gap is often only achieved with massive exploration of the state-space. In case of linear systems, we discuss how a relatively small but sufficiently rich dataset can be exploited to generate new constraints offline and without observing the corresponding transitions. Moreover, we show how to reconstruct the associated unknown stage-costs and, when the system is stochastic, we offer insights on the related problem of estimating the expected value in the Bellman operator without re-initializing the dynamics in the same state-input pairs.

I. INTRODUCTION

The linear programming (LP) approach to optimal control problems was initially developed by A.S. Manne in the 1960s [15], following the well-known studies conducted by R. Bellman in the 1950s [2]. The idea is to exploit the monotonicity and contractivity properties of the Bellman operator [3] to build LPs whose solution is the optimal value function. An evident advantage of the LP formulation is that there exist efficient and fast algorithms to tackle such programs [7]. On the other hand, similarly to the classic dynamic programming approach introduced by Bellman, the LP approach suffers from poor scalability properties referred to as *curse of dimensionality* [5]. The sources of intractability for systems with continuous state and action spaces can be identified as, among others, an optimization variable in an infinite dimensional space and an infinite number of constraints. For this reason the infinite dimensional LPs are usually approximated by tractable finite dimensional ones [17], [11], [19], [6].

In recent years, the LP approach has experienced an increasing interest, especially in combination with model-free control techniques [20], [1], [16], [23]. In such a setting, one assumes the dynamical system to be unknown but observable via state-space exploration, and builds one Bellman inequality (or constraint) of the LP for each observed transition. In this way, it is possible to both bypass the more classic system identification step and mitigate a source of intractability by solving an LP with a finite amount of constraints. A discussion on the approximation introduced by constraint sampling can be found in [12]. Both theoretical

and empirical evidence present in the previously mentioned literature suggest that a massive amount of data is generally needed to comply with a desired performance level. As discussed in the scalability analysis performed in [16], this aspect becomes even more evident for large-scale systems. Moreover, another relevant problem in the LP approach is to provide an estimate for the expected value in the constraints. This is usually performed by re-initializing the dynamics in the same state-input pairs and computing a Monte Carlo estimate of the associated value function evaluated at the next state [20]. In a stochastic framework, unfortunately, it may be practically impossible to re-initialize the system at desired states.

Another thriving data-driven research direction is the one revolving around behavioural theory and Willem's *fundamental lemma* [28], stating that the information contained in a sufficiently long trajectory of a linear system is, under mild assumptions, enough to describe any other trajectory of the same length that can be generated by the system itself. In a model-free context, a so-called *persistently exciting* exploration input is often used to generate such trajectories, obtain a data-based representation of the underlying linear system and develop control techniques such as MPC [8] or assess system's properties such as dissipativity [18] and stabilizability/controllability [13]. The authors in [25] discuss the conditions for which a dataset is informative, *i.e.* when the data contain enough information to accomplish a specific control task.

Motivated by the poor scalability often affecting the LP approach and inspired by the recent literature on data-driven control of linear systems, in the present work we discuss how to mitigate the cost of performing massive exploration. After introducing the problem general formulation in Sec. II, our main contributions can be summarised as follows:

- We show in Sec. III-A that a sufficiently rich dataset can be used to generate all the constraints involved in the LP formulation for a linear system, offline and without observing the corresponding transitions;
- Moreover, thanks to a bilinear algebra framework, we show in Sec. III-B how to reconstruct the associated stage-cost evaluations starting again from a fixed dataset;
- For stochastic systems, we provide in Sec. IV insights on the estimation of the expected values in the constraints of the LP, without resorting to iterative dynamics re-initialization.

We denote with $\mathbb{M}_p$ the set of $p \times p$ real matrices and with $\tilde{\mathbb{M}}_p \subset \mathbb{M}_p$ the subset of symmetric matrices. A vector of ones of suitable dimension is denoted with $\mathbf{1}$. The vectorization of a symmetric matrix $M \in \tilde{\mathbb{M}}_p$ with entries $[M]_{ij} = m_{ij}$ is $\text{vec}(M) = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{pp} \end{bmatrix} \in \mathbb{R}^{p^2}$ and its trace is denoted with $\text{tr}(M)$. A symmetric bilinear form [22] is a map $\beta : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ that is linear in its arguments taken separately, and such that $\beta(x, y) = \beta(y, x)$ for all $x, y \in \mathbb{R}^p$. A pair $(\mathbb{R}^p, \beta)$ defines a bilinear space. We also define the quadratic form $\ell : \mathbb{R}^p \to \mathbb{R}$ associated to β as $\ell(z) = \beta(z, z)$ for all $z \in \mathbb{R}^p$. The following holds:

$$\ell(az) = a^2 \ell(z) \quad \forall a \in \mathbb{R}, \; \forall z \in \mathbb{R}^p \tag{1}$$

$$\beta(x, y) = \tfrac{1}{2}(\ell(x+y) - \ell(x) - \ell(y)) \quad \forall x, y \in \mathbb{R}^p. \tag{2}$$

Moreover, given a basis $\mathcal{B} = \{b_1, \ldots, b_p\}$ for $\mathbb{R}^p$, we denote with $[M^{\mathcal{B}}]_{ij} = \beta(b_i, b_j)$ the matrix representation of β in the basis $\mathcal{B}$.

II. OPTIMAL CONTROL VIA LINEAR PROGRAMMING

Consider a discrete-time stochastic dynamical system

$$x_{k+1} = f(x_k, u_k, \xi_k), \tag{3}$$

with (possibly infinite) state and action spaces $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$. Here, $\xi_k \in \mathbb{R}^n$ denotes the realizations of independent identically distributed (i.i.d.) random variables, and $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ is the map encoding the dynamics. We consider *stationary feedback policies*, given by functions $\pi : \mathbb{R}^n \to \mathbb{R}^m$; for more general classes of policies, see [14]. A nonnegative cost is associated to each state-action pair through the *stage cost* function $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$. We introduce a *discount factor* $\gamma \in (0, 1)$ and consider the infinite-horizon cost associated to policy π

$$v_\pi(x) = \mathbb{E}_\xi \left[\sum_{k=0}^{\infty} \gamma^k \ell(x_k, \pi(x_k)) \;\middle|\; x_0 = x \right]. \tag{4}$$

The objective of the optimal control problem is to find an optimal policy π^* such that $v_{\pi^*}(x) = \inf_\pi v_\pi(x) = v^*(x)$, where v^* is known as the optimal *value function*. Let us define the vector space of all real-valued measurable functions that have a finite r-weighted sup-norm [4, §2.1] as

$$\mathbb{V} = \{ v : \mathbb{R}^n \to \mathbb{R} \;\mid\; ||v||_{\infty, r} < \infty \}. \tag{5}$$

Throughout the paper, we work under [14, Assump. 4.2.1 and 4.2.2] to ensure that $v^* \in \mathbb{V}$, π^* is measurable and the infimum of v_π is attained. The optimal value function can be expressed as the solution of the following infinite-dimensional linear program [14], [11]

$$\sup_{v \in \mathbb{V}} \int_{\mathbb{R}^n} v(x) c(dx)$$
$$\text{s.t. } v(x) \leq \ell(x, u) + \gamma \mathbb{E}_\xi v(f(x, u, \xi)) \quad \forall (x, u), \tag{6}$$

where c is a finite measure that assigns positive mass to all open subsets of $\mathbb{R}^n$. The above formulation is not solvable in general due to several sources of intractability, see *e.g.* [6]

and [26]. If one is nonetheless able to obtain v^*, they can in principle compute the corresponding policy by

$$\pi^*(x) = \arg\min_u \{ \ell(x, u) + \gamma \mathbb{E}_\xi v^*(f(x, u, \xi)) \}. \tag{7}$$

A special case of the infinite-horizon optimal control problem arises when the dynamics is linear

$$f(x, u, \xi) = Ax + Bu + \xi, \tag{8}$$

with $A \in \mathbb{M}_n, B \in \mathbb{R}^{n \times m}$, and the cost function is quadratic

$$\ell(x, u) = \begin{bmatrix} x \\ u \end{bmatrix}^\mathsf{T} L \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} x \\ u \end{bmatrix}^\mathsf{T} \begin{bmatrix} L_{xx} & L_{xu} \\ L_{xu}^\mathsf{T} & L_{uu} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}. \tag{9}$$

For such linear-quadratic (LQ) problems we impose the following assumption.

Assumption 1: The pair (A, B) is stabilizable, ξ is i.i.d. with zero mean and covariance matrix Σ. Moreover, $L \succeq 0$ and $L_{uu} \succ 0$.

By denoting $P \in \tilde{\mathbb{M}}_n$ and $e \in \mathbb{R}^n$, let us define

$$\mathbb{V}_q = \{ v : \mathbb{R}^n \to \mathbb{R} \;\mid\; v(x) = x^\mathsf{T} P x + e \} \subset \mathbb{V}. \tag{10}$$

The solution to the LP (6) under LQ assumptions (8)-(9) is then $v^*(x) = x^\mathsf{T} P^* x + e^* \in \mathbb{V}_q$, where P^* is the solution to the well-known associated algebraic Riccati equation (ARE) [10] and $e^* = \frac{\gamma}{1-\gamma} \text{tr}(P^* \Sigma)$. By imposing c to be a probability distribution with zero mean and identity covariance matrix and restricting $v \in \mathbb{V}_q$, an equivalent formulation for (6) that directly involves P and e is [6]

$$\begin{aligned} \max_{P, e} \quad & \text{tr}(P) + e \\ \text{s.t.} \quad & x^\mathsf{T} P x \leq \ell(x, u) \\ & + \gamma \mathbb{E}_\xi (Ax + Bu + \xi)^\mathsf{T} P(Ax + Bu + \xi), \end{aligned} \tag{11}$$

for all (x, u).

In model-based control, typically one assumes that f is known and directly looks for solutions to, *e.g.*, the ARE or finite-dimensional semidefinite programs using convex optimisation tools. In model-free control on the other hand, one assumes that the model is unknown but can be observed indirectly through data (x^i, u^i, x^{i+}) of state-input pairs $(x^i, u^i) \in \mathbb{R}^n \times \mathbb{R}^m$ and the next state $x^{i+} = f(x^i, u^i, \xi^i)$. The solution to (4) can be estimated, *e.g.*, by means of reinforcement learning methods [21]. Moreover, one can also obtain the optimal policy by reformulating (6)-(7) in terms of the so-called Q-function [27]. To keep the discussion simple, however, we do not explore this direction here. In the context of the LP approach, the infinite constraints needed to construct the feasible region in (6) or (11) are often replaced with a finite subset, each one associated with a data tuple (x^i, u^i, x^{i+}), as argued in [1], [12], [16], [23].

III. FEASIBLE REGION SYNTHESIS FROM DATA

A. Unknown dynamics

The information contained in a sufficiently long trajectory of a linear system is, under mild assumptions, enough to describe any other trajectory of the same length that can be generated by the system [28]. In this section we discuss

the consequences of this idea and adapt the theoretical implications to the context of the data-driven LP approach. Throughout the section we work under LQ assumptions (8)-(9), Assumption 1, and we consider deterministic dynamics, *i.e.* $\xi = 0$ for all time steps and $\Sigma = 0$. First, we provide a description in matrix form of the Bellman inequalities.

Proposition 1: The constraint set in (11) is equivalent to

$$\text{vec}(H(x,u))^\intercal \text{vec}(P) \le \ell(x,u) \quad \forall(x,u), \qquad (12)$$

where $H : \mathbb{R}^n \times \mathbb{R}^m \to \tilde{\mathbb{M}}_n$ is

$$H(x,u) = xx^\intercal - \gamma(Ax + Bu)(Ax + Bu)^\intercal. \qquad (13)$$

Proof: We can express the constraint set in (11) as

$$\sum_{i=1}^n \sum_{j=i}^n h_{ij}(x,u)p_{ij} \le \ell(x,u) \quad \forall(x,u), \qquad (14)$$

where

$$h_{ij}(x,u) = x_i x_j - \gamma(Ax + Bu)_i (Ax + Bu)_j.$$

Then by imposing $[H(x,u)]_{ij} = h_{ij}(x,u)$ we obtain (13) and, finally, we can re-arrange the left-hand side of (14) in vector form and obtain (12). ∎

We say that (X,U,X^+) is a dataset of length T when $X = \begin{bmatrix} x^1 & \cdots & x^T \end{bmatrix}$, $U = \begin{bmatrix} u^1 & \cdots & u^T \end{bmatrix}$ and $X^+ = AX + BU$. We also introduce the following assumption.

Assumption 2: $\text{rank} \begin{bmatrix} X \\ U \end{bmatrix} = n + m.$

The following lemma shows how all the infinite constraints in (11) can potentially be reconstructed offline directly from the dataset without explicitly determining the matrices A and B and without observing the corresponding system's transitions.

Lemma 1: Consider a dataset (X,U,X^+) of length T satisfying Assumption 2. Then, for each $(x,u) \in \mathbb{R}^n \times \mathbb{R}^m$ there exists an $\alpha \in \mathbb{R}^T$ and such that

$$H(x,u) = (X\alpha)(X\alpha)^\intercal - \gamma(X^+\alpha)(X^+\alpha)^\intercal. \qquad (15)$$

Proof: Since $\begin{bmatrix} X^\intercal & U^\intercal \end{bmatrix}^\intercal$ is full row-rank by assumption, we know there exists an α satisfying

$$\begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} X \\ U \end{bmatrix} \alpha. \qquad (16)$$

Moreover, since

$$AX\alpha + BU\alpha = (AX + BU)\alpha = X^+\alpha, \qquad (17)$$

we have that

$$\begin{aligned} H(x,u) &= xx^\intercal - \gamma(Ax + Bu)(Ax + Bu)^\intercal \\ &= (X\alpha)(X\alpha)^\intercal - \gamma(X^+\alpha)(X^+\alpha)^\intercal, \end{aligned}$$

that concludes the proof. ∎

Thanks to Proposition 1 and Lemma 1 we know that, given a dataset (X,U,X^+) satisfying Assumption 2, we can reconstruct each of the infinite constraints in (11) by computing $H(x,u)$ at suitable linear combinations of X and X^+. Figures 1 and 2 illustrate the fundamental role of the sampled constraints for the suboptimality of the derived solution and,
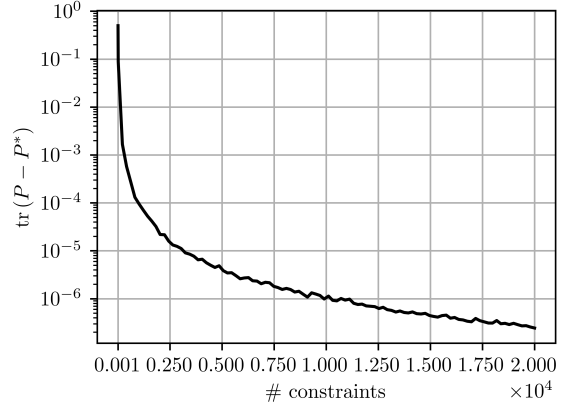

Fig. 1: Median across 10^3 independent runs of the optimality gap versus the number of constraints for system (18).

consequently, the utility of the proposed artificial sampling technique on the linear system

$$x^+ = \begin{bmatrix} 1 & 0.1 \\ 0.5 & -0.5 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u. \qquad (18)$$

In particular, only the first 10 constraints are generated via simulations of the system. The collected data are then used to synthesise new constraints according to (15) for randomly selected values of α. As depicted in Fig. 1, the additional constraints allow for a dramatic improvement of the optimality gap. A graphical representation of the support constraints displacement is plotted in Fig 2a in the variables space (p_{11}, p_{12}, p_{22}). In particular, we first solve the LP with the 10 observed constraints (blue dot), and then we solve the LP again by including 10 additional constraints generated artificially (red dot). Fig. 2b shows the corresponding improvement in the value function. As observed experimentally, in general generating enough constraints from exploration to reach a prescribed performance level can be prohibitive in a real scenario. Our proposed approach alleviates this issue by allowing one to only sample a small subset of the state-space and then inexpensively synthesise new constraints offline.

Artificial constraints generation can also be exploited in a policy iteration (PI) fashion [3] to, *e.g.*, complement the approach proposed in [1]. The initialization of the data-driven PI algorithm can be performed by exploring the state-space, as described in [1]. Then, at any successive step t of the algorithm one can emulate the PI behaviour by selecting appropriate vectors α that target state-input pairs associated with the current policy as follows

$$\begin{bmatrix} x \\ K_t x \end{bmatrix} = \begin{bmatrix} X \\ U \end{bmatrix} \alpha, \qquad (19)$$

without need for further exploration.

B. Unknown stage cost

In the context of control applications, the stage-cost is formulated by the designer: it is thus reasonable to consider $\ell(x,u)$ to be known. In cases where the stage-cost

is not known, the following proposition provides a way to reconstruct $\ell(x, u)$ if we observe the cost incurred at a finite number of state-input pairs.

Proposition 2: Consider a dataset (X, U, X^+) satisfying Assumption 2, and the square matrix $\begin{bmatrix} \tilde{X}^\intercal & \tilde{U}^\intercal \end{bmatrix}^\intercal \in \mathbb{M}_{n+m}$ obtained by down-selecting $n+m$ columns from $\begin{bmatrix} X^\intercal & U^\intercal \end{bmatrix}^\intercal$ such that Assumption 2 is still satisfied. Then, for each $(x, u) \in \mathbb{R}^n \times \mathbb{R}^m$ there exists an $\alpha \in \mathbb{R}^{n+m}$ such that

$$\ell(x, u) = \alpha^\intercal L_{X,U} \alpha, \tag{20}$$

where $L_{X,U} \in \tilde{\mathbb{M}}_{n+m}$ is

$$[L_{X,U}]_{ij} = \beta\left(\begin{bmatrix} x^i \\ u^i \end{bmatrix}, \begin{bmatrix} x^j \\ u^j \end{bmatrix} \right), \tag{21}$$

and β is the bilinear form associated to ℓ.

Proof: First note that there exists a unique α satisfying

$$\begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} \tilde{X} \\ \tilde{U} \end{bmatrix} \alpha. \tag{22}$$

Let us temporarily denote $z = \begin{bmatrix} x & u \end{bmatrix}^\intercal$ and $Z = \begin{bmatrix} \tilde{X}^\intercal & \tilde{U}^\intercal \end{bmatrix}^\intercal$ such that z^i is the i-th column of Z and α^i is the i-th entry of α. Moreover, we recall that, since $\ell : \mathbb{R}^{n+m} \to \mathbb{R}$ is a quadratic form, properties (1)-(2) hold. This allows us to express $\ell(x, u) = \ell(z) = \ell(Z\alpha)$ as

$$\ell(Z\alpha) = \ell\left(\sum_{i=1}^{n+m} \alpha^i z^i \right)$$
$$= \ell(\alpha^1 z^1) + \ell\left(\sum_{i=2}^{n+m} \alpha^i z^i \right) + 2\beta\left(\alpha^1 z^1, \sum_{i=2}^{n+m} \alpha^i z^i \right).$$

On the other hand, it also holds

$$\ell\left(\sum_{i=2}^{n+m} \alpha^i z^i \right) = \ell(\alpha^2 z^2) + \ell\left(\sum_{i=3}^{n+m} \alpha^i z^i \right) + 2\beta\left(\alpha^2 z^2, \sum_{i=3}^{n+m} \alpha^i z^i \right).$$
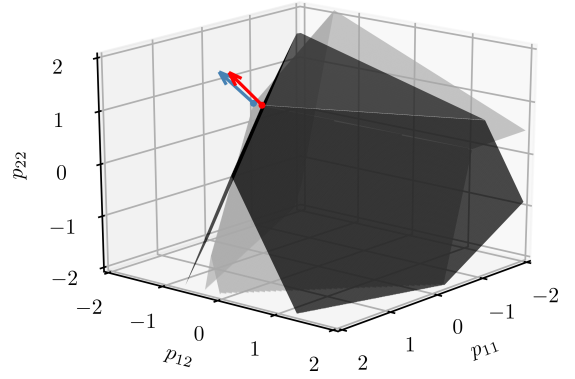
Hence, $\ell(Z\alpha)$ can be written recursively as

$$\ell(Z\alpha) = \sum_{i=1}^{n+m} \ell(\alpha^i z^i) + 2\sum_{i=1}^{n+m} \beta\left(\alpha^i z^i, \sum_{j=i+1}^{n+m} a^j z^j \right)$$
$$= \sum_{i=1}^{n+m} \alpha^{i^2} \ell(z^i) + 2\sum_{i=1}^{n+m} \sum_{j=i+1}^{n+m} \alpha^i \alpha^j \beta(z^i, z^j)$$
$$= \alpha^\intercal \begin{bmatrix} \ell(z^1) & \beta(z^1, z^2) & \cdots & \beta(z^1, z^k) \\ \star & \ell(z^2) & \cdots & \beta(z^2, z^k) \\ \star & \star & \ddots & \vdots \\ \star & \star & \star & \ell(z^k) \end{bmatrix} \alpha$$
$$= \alpha^\intercal L_{X,U} \alpha, \tag{23}$$

where the symbol $\star$ denotes symmetry. Finally, since $\ell(z^i) = \beta(z^i, z^i)$, we obtain (20)-(21). ∎
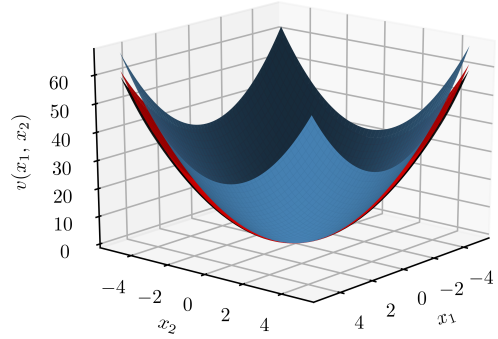
Note that if we express an arbitrary state-input pair (x, u) as a linear combination of our data, as in (22), we can write

$$\ell(x, u) = \begin{bmatrix} x \\ u \end{bmatrix}^\intercal L \begin{bmatrix} x \\ u \end{bmatrix} = \alpha^\intercal \underbrace{\begin{bmatrix} \tilde{X} \\ \tilde{U} \end{bmatrix}^\intercal L \begin{bmatrix} \tilde{X} \\ \tilde{U} \end{bmatrix}}_{L_{X,U}} \alpha.$$

It becomes evident that L and $L_{X,U}$ are congruent matrices and therefore they are two matrix representations of the same

(a) Relative displacement of the support constraints generated with exploration (light grey) and artificial sampling (dark grey) for the LP associated with system (18). The colored dots and arrows represent the optimal solutions of the LPs and the gradient direction, respectively.

(b) Color-matched graphical representation of the quadratics associated with the solutions depicted in Fig. 2a and the optimal value function (in black).

Fig. 2: Graphical comparison of the support constraints and solutions for the LPs associated with system (18) when different sets of constraints are considered.

quadratic form ℓ expressed in two different bases [22]. Under this light, the data matrix $\begin{bmatrix} \tilde{X}^\intercal & \tilde{U}^\intercal \end{bmatrix}^\intercal$ takes the role of the matrix transforming the basis of L into the basis of $L_{X,U}$.

Regarding the computation of $L_{X,U}$, according to (23) and for each entry (i, j) of $L_{X,U}$, we have to evaluate β at the corresponding $(x^i, u^i), (x^j, u^j)$ picked from our dataset. We already have the $n+m$ diagonal terms of $L_{X,U}$ as they are direct stage cost evaluations $\ell(x^i, u^i)$. As for the off-diagonal terms, by recalling once again Equation (2), we still have to add $\binom{n+m}{2}$ observations in our dataset, one for each pairwise combination of z^i and z^j. The total amount of observations needed to compute $L_{X,U}$ is $n+m+\binom{n+m}{2} = \frac{(n+m)(n+m+1)}{2}$ that, expectedly, equals the amount of unknown entries in L.

C. State-space exploration

A dataset (X, U, X^+) satisfying Assumption 2 can be generated by initializing the dynamics at desired states and applying the suitable inputs to ensure the rank condition is satisfied. In case targeted initialization is not possible, one can build independent samples by initializing the dynamics at an arbitrary state and running a long and rich enough exploration sequence, often guaranteed by the *persistence*

of excitation condition on the input [28], [13]. In detail, a sequence $u^1, \ldots, u^T \in \mathbb{R}^m$ is said to be persistently exciting of order L if the associated Hankel matrix of depth L,

$$\mathcal{H}_L = \begin{bmatrix} u^1 & u^2 & \cdots & u^{T-L+1} \\ u^2 & u^3 & \cdots & u^{T-L+2} \\ \vdots & \vdots & & \vdots \\ u^L & u^{L+1} & \cdots & u^T \end{bmatrix} \in \mathbb{R}^{(mL) \times (T-L+1)}, \quad (24)$$

has full row rank mL. It is evident that such condition can only be satisfied if $T \geq L(m+1) - 1$. Consider then to excite the system $x^+ = Ax + Bu$ with a persistently exciting sequence $u^1, \ldots, u^T \in \mathbb{R}^m$ of order $n+1$, implying that $T \geq n(m+1)+m$, and record the associated state transitions $x^1, \ldots, x^T \in \mathbb{R}^n$. Then, [28, Corollary 2.(ii)] ensures that

$$\mathrm{rank} \begin{bmatrix} x^1 & \cdots & x^T \\ u^1 & \cdots & u^T \end{bmatrix} = n + m, \quad (25)$$

and Assumption 2 is satisfied, as discussed in [13].

Between targeted initialization and a single exploration sequence, we can mention the works in [24] and [9] where condition (25) is guaranteed even when the dataset is composed by multiple (possibly short) roll-outs.

As mentioned in the introduction, in recent literature on data-driven control Willem's *fundamental lemma* is often used to obtain a data-based representation of the trajectory space of a linear system and develop analysis and control techniques. Within the context of the LP approach, we show how to construct all infinite constraints compatible with system's dynamics starting from a sufficiently rich dataset, allowing one to avoid massive sampling.

IV. FEASIBLE REGION ESTIMATION FOR STOCHASTIC SYSTEMS

In the context of the LP approach, a fundamental issue when dealing with stochastic systems is the estimation of the expected values in the Bellman inequalities. As discussed *e.g.* in [20] and [16], one could re-initialize the dynamics at a fixed state-input pair (x, u) a sufficient number of times N, observe the corresponding transition $f(x, u, \xi)$ and estimate $\mathbb{E}_\xi[v(f(x, u, \xi))]$ by averaging the observations in a Monte Carlo fashion, as

$$\frac{1}{N} \sum_{k=1}^N v(f(x, u, \xi^k)) \approx \mathbb{E}_\xi[v(f(x, u, \xi))]. \quad (26)$$

On the other hand, such an estimation can only be performed if one can re-initialize the dynamics at the same state x and play the same input u multiple times. Clearly this assumption is limiting in a stochastic framework, since it may be impossible to re-initialize the system at a desired state in general. Here we discuss the effect of removing the re-initialization assumption by averaging the observations over the next state $f(x, u, \xi)$ instead of over the value function $v(f(x, u, \xi))$ under LQ assumptions (8)-(9).

First, we give a matrix description of the Bellman inequalities by specializing Proposition 1 to stochastic linear systems.

Proposition 3: The constraint set in (11) is equivalent to

$$\mathrm{vec}(\mathbb{E}_\xi G(x, u, \xi))^{\mathsf{T}} \mathrm{vec}(P) \leq \ell(x, u) \quad \forall (x, u), \quad (27)$$

where $G : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \tilde{\mathbb{M}}_n$ is

$$G(x, u, \xi) = xx^{\mathsf{T}} - \gamma(Ax + Bu + \xi)(Ax + Bu + \xi)^{\mathsf{T}}. \quad (28)$$

Proof: The result holds by considering a similar reasoning to the one in the proof of Proposition 1. ∎

As expected, note that in case of deterministic dynamics (27) reduces to (12), as $G(x, u, 0) = H(x, u)$. For zero mean noise, the expectation of G is given by

$$\mathbb{E}_\xi G(x, u, \xi) = H(x, u) - \gamma \Sigma, \quad (29)$$

and the effect of the noise boils down to a constant term in the matrix of coefficients.

In the context of the data-driven LP approach, one could be tempted to directly use data observed from the evolution of the stochastic dynamics, constructing a set of noise-corrupted constraints of the form

$$\mathrm{vec}(G(x, u, \xi))^{\mathsf{T}} \mathrm{vec}(P) \leq \ell(x, u). \quad (30)$$

Alternatively, a Monte Carlo approach could be employed to mitigate the effect of the noise. In the linear context, the estimation with re-initialization (26) corresponds to performing the following approximation

$$\frac{1}{N} \sum_{k=1}^N G(x, u, \xi^k) \approx \mathbb{E}_\xi G(x, u, \xi). \quad (31)$$

In this case, however, re-initialization can be circumvented by averaging the x^+ directly instead of G. Consider to have a dataset (X, U, X^+) of sufficient length N, where $\bar{x} = \frac{1}{N} X \mathbf{1}$ and $\bar{u} = \frac{1}{N} U \mathbf{1}$ are the average state and input while $\bar{x}^+ = \frac{1}{N} X^+ \mathbf{1}$ is the observed sample mean of the transition over the N realizations included in the dataset.

Proposition 4: Consider system (8) and $v \in \mathbb{V}_q$. Then,

$$G(\bar{x}, \bar{u}, \bar{\xi}) = \bar{x}\bar{x}^{\mathsf{T}} - \gamma \bar{x}^+ \bar{x}^{+\mathsf{T}}, \quad (32)$$

where $\bar{\xi} = \frac{1}{N} \sum_{k=1}^N \xi^k$ is the sample mean of the noise over the corresponding N realizations.

Proof: Thanks to the linearity of the dynamics it holds

$$\bar{x}^+ = \frac{1}{N} \sum_{k=1}^N (Ax^k + Bu^k + \xi^k) = A\bar{x} + B\bar{u} + \bar{\xi},$$

hence

$$\begin{aligned} G(\bar{x}, \bar{u}, \bar{\xi}) &= \bar{x}\bar{x}^{\mathsf{T}} - \gamma(A\bar{x} + B\bar{u} + \bar{\xi})(A\bar{x} + B\bar{u} + \bar{\xi})^{\mathsf{T}} \\ &= \bar{x}\bar{x}^{\mathsf{T}} - \gamma \bar{x}^+ \bar{x}^{+\mathsf{T}}, \end{aligned}$$

concluding the proof. ∎

Note that in order to compute $G(\bar{x}, \bar{u}, \bar{\xi})$ as in (32) we do not need to know $\bar{\xi}$ which is, as a matter of fact, unknown and embedded in the dynamics; we only need to compute $\bar{x}$ and $\bar{x}^+$ from our dataset. We also stress that (32) holds for any dataset irrespective of Assumption 2. Therefore, we can substitute approximation (31) with the following

$$G(x, u, \bar{\xi}) \approx G(x, u, 0) = H(x, u), \quad (33)$$

where the upper bar notation has been removed to stress the fact that (32) can in principle be computed for arbitrary (x, u) depending on the available data.

According to equation (29), we are neglecting the contribution of the covariance matrix into the constraints, ending up solving a sampled version of the LP for the deterministic system $x^+ = Ax + Bu$. It is well-known [10] that the difference between the optimal value function for a stochastic linear system and its corresponding deterministic one is a constant shift depending on the covariance matrix. Consequently, the two associated policies (see Eq. (7)) coincide.

Therefore, we circumvented the re-initialization condition at the expense of approximating a value function that, asymptotically in the number of sampled constraints, is the one associated with the deterministic dynamics. In case one is interested in policy search only, this heuristic could represent a viable choice since it asymptotically preserves the optimal policy. In general, on the other hand, we want to stress that computing the associated policy with (7) requires knowledge of f. For this reason, in order to make this heuristic operational, one should first reformulate the LP in terms of Q-functions, such that the policy extraction does not depend on the matrices A and B.

Finally, we discuss a heuristic on constructing approximated artificial constraints. Consider to have a dataset of length NT and to partition the data into T subsets (X^i, U^i, X^{i+}) of length N, so that $X^{i+} = AX^i + BU^i + D^i$ and $D^i = [\xi^{i1} \ \dots \ \xi^{iN}]^\intercal$ contains the corresponding noise realizations. No rank assumption is needed on any of the T datasets. Then, compute the average dataset $(\bar{X}, \bar{U}, \bar{X}^+)$, where $\bar{X} = [\bar{x}^1 \ \dots \ \bar{x}^N]^\intercal$, $\bar{U} = [\bar{u}^1 \ \dots \ \bar{u}^N]^\intercal$, $\bar{X}^+ = [\bar{x}^{+1} \ \dots \ \bar{x}^{+N}]^\intercal$, and each of their columns is $\bar{x}^i = \frac{1}{N}X^i\mathbf{1}$, $\bar{u}^i = \frac{1}{N}U^i\mathbf{1}$ and $\bar{x}^{+i} = \frac{1}{N}X^{+i}\mathbf{1}$. Since the partition into datasets is arbitrary it is reasonable to consider that Assumption 2 can be satisfied for $(\bar{X}, \bar{U}, \bar{X}^+)$. Note that, for N sufficiently large, we can approximate $\bar{X}^+ \approx A\bar{X} + B\bar{U}$. As a consequence, we can exploit (15) to artificially build a desired number of (approximated) constraints associated with the deterministic system $x^+ = Ax + Bu$. Further discussion is necessary to provide probabilistic performance bounds on the error introduced and it is deferred to future studies.

V. CONCLUSIONS AND FUTURE WORK

On the wave of the exciting recent literature on behavioural theory, we showed how to synthesise new constraints for the LP formulation of a linear system starting from a suitable dataset. In this way, the often poor scalability properties of the LP approach are partially alleviated by generating constraints offline and without observing the dynamics evolution. Other significant insights were given about reconstructing the associated unknown stage-costs.

Many important issues are still to be explored and discussed, such as extending the approach to the Q-function formulation and relaxing the linearity assumptions on the dynamics to affine or polynomial.

REFERENCES

[1] G. Banjac and J. Lygeros. A data-driven policy iteration scheme based on linear programming. In *58th IEEE Conference on Decision and Control*, pages 816–821, 2019.

[2] R. Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*, 38(8):716–719, 1952.

[3] D.P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II.* Athena Scientific, 3rd edition, 2007.

[4] D.P. Bertsekas. *Abstract Dynamic Programming.* Athena Scientific, 2013.

[5] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1st edition, 1996.

[6] P.N. Beuchat, A. Georghiou, and J. Lygeros. Performance guarantees for model-based approximate dynamic programming in continuous spaces. *IEEE Transactions on Automatic Control*, 65(1):143–158, 2020.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[8] J. Coulson, J. Lygeros, and F. Dörfler. Data-enabled predictive control: In the shallows of the deepc. In *18th European Control Conference (ECC)*, pages 307–312, 2019.

[9] J. Coulson, J. Lygeros, and F. Dörfler. Distributionally robust chance constrained data-enabled predictive control, 2020. ArXiv2006.01702.

[10] M.H.A. Davis and R.B. Vinter. *Stochastic Modelling and Control.* Chapman and Hall, 1985.

[11] D.P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

[12] D.P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

[13] C. De Persis and P. Tesi. Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924, 2020.

[14] O. Hernandez-Lerma and J.B. Lasserre. *Discrete-Time Markov Control Processes: Basic Optimality Criteria.* Springer-Verlag NY, 1996.

[15] A.S. Manne. Linear programming and sequential decisions. *Management Science*, 6(3):259–267, 1960.

[16] A. Martinelli, M. Gargiani, and J. Lygeros. Data-driven optimal control with a relaxed linear program. 2020. arXiv:2003.08721.

[17] P. Mohajerin Esfahani, T. Sutter, D. Kuhn, and J. Lygeros. From infinite to finite programs: Explicit error bounds with applications to approximate dynamic programming. *SIAM Journal on Optimization*, 28(3):1968–1998, 2018.

[18] A. Romer, J. Berberich, J. Köhler, and F. Allgöwer. One-shot verification of dissipativity properties from input–output data. *IEEE Control Systems Letters*, 3(3):709–714, 2019.

[19] P.J. Schweitzer and A. Seidmann. Generalized polynomial approximations in markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, 1985.

[20] T. Sutter, A. Kamoutsi, P. Mohajerin Esfahani, and J. Lygeros. Data-driven approximate dynamic programming: A linear programming approach. In *56th IEEE Conference on Decision and Control*, pages 5174–5179, 2017.

[21] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction.* The MIT Press, 2017.

[22] K. Szymiczek. *Bilinear algebra: An introduction to the algebraic theory of quadratic forms.* Gordon&Breach Science Publishers, 1997.

[23] A. Tanzanakis and J. Lygeros. Data-driven control of unknown systems: A linear programming approach. In *IFAC-PapersOnLine*, 2020. arXiv:2003.00779.

[24] H.J. van Waarde, C. De Persis, M.K. Camlibel, and P. Tesi. Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Systems Letters*, 4(3):602–607, 2020.

[25] H.J. van Waarde, J. Eising, H.L. Trentelman, and M.K. Camlibel. Data informativity: A new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, 65(11):4753–4768, 2020.

[26] Y. Wang, B. O'Donoghue, and S. Boyd. Approximate dynamic programming via iterated Bellman inequalities. *International Journal of Robust and Nonlinear Control*, 25(10):1472–1496, 2015.

[27] C.J.C.H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, May 1992.

[28] J.C. Willems, P. Rapisarda, I. Markovsky, and B.L.M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.