An update-and-design scheme for scenario-based LQR synthesis

Anna Scampicchio, Andrea Iannelli

***Abstract*— This paper deals with a finite-horizon Linear Quadratic Regulator (LQR) design for unknown linear time-invariant plants. The objective is to provide a flexible approach which gives robustness guarantees on the closed-loop cost, while avoiding an overly conservative design. The proposed method consists of computing the posterior distribution of the system's matrices, and using samples from the desired credible region to solve a convex scenario-based program; the result is an open-loop solution of the robust LQR that is then expressed as state-feedback and is used to obtain a new system trajectory. By updating the system estimates as more data are gathered, the algorithm ensures controllers the same robustness guarantees while having to cope with less dispersion in the samples.**

I. INTRODUCTION

The problem of controlling unknown plants has become central to many engineering applications, as a result of the increasing complexity of modern systems. The *indirect* approach to address this problem consists of first identifying a dynamic model of the system from available data [1], [2] and then designing a suitable controller using model-based techniques. Alternatively, one can seek a *direct* map from the system's trajectories to the controller's actions, by-passing the identification of the plant's dynamics. This approach has found particular success in the reinforcement learning community [3]. Determining the best strategy is often a problem-dependent matter, since the respective approaches offer distinct advantages and disadvantages: see, e.g., [4], [5], [6] for related surveys, and the investigation in [7] through the lens of data-driven control.

In this work, emphasis is put on two crucial and inter-related aspects arising when controlling unknown plants: robustness and conservatism of the design. Robustness is a multifaceted concept and we consider it here in the standard robust control sense, namely as the requirement to optimize performance over all possible disturbances to the plant that are compatible with its current knowledge [8]. In indirect approaches, this is typically achieved by using a fixed uncertainty region centered around an estimate of the model, and thus it might incur conservatism if the initial knowledge of the system is not precise [9], [10]. In direct approaches, conservatism is less of an issue, since new data are constantly used to make decisions, but it is much more challenging to guarantee robustness. The aim of this work is thus to formulate a control design scheme that, on the one hand, allows for robustness guarantees, and on the other can leverage new observations from the system's response to decrease its conservatism.

Anna Scampicchio is a member of the Institute for Dynamic Systems and Control, ETHZ, Zürich 8092, Switzerland (ascampicc@ethz.ch). Andrea Iannelli is a member of the Department of Electrical Engineering, Automatic Control Lab, ETHZ, Zürich 8092, Switzerland (iannelli@control.ee.ethz.ch).

The problem under investigation is the finite-horizon, discrete-time Linear Quadratic Regulator (LQR) design, which is key in optimal control theory [11], is widely used in practice and is still the subject of ongoing research: see, e.g., [12], [13], [14], [15], [16], [17], [18], [19], [20]. We focus on the framework of indirect control design and propose a method that leverages Bayesian estimation of the unknown system matrices. An uncertainty region of probability $\varpi \in (0, 1)$ is then extracted from such a distribution and is used to solve a sample-based, worst-case LQR program. Robustness is achieved by leveraging the scenario-approach framework [21]. Specifically, we build on [22] for this part, where a scenario program was formulated to minimize the LQR cost associated with a certain number of samples coming from the distribution of the unknown system's matrices. There, the set-up was distribution-agnostic; here, samples are actively drawn from the current posterior distribution, which is iteratively updated with new data. As a consequence, the size of its uncertainty region is shrinking. Crucially, since the scenario-approach guarantees that the generalization capability of the solution only depends on the number of scenarios and the number of optimization variables (which do not change throughout the iterations), the proposed approach provides a controller with the same robustness properties as the data-agnostic sample-driven solution, but with decreased conservatism. It is also observed that, even though this is by nature an indirect approach, because an estimate of the system is required, it is conceptually different than standard robust control methods, as the design is not done on a nominal plus uncertainty estimate of the model (e.g., via S-Lemma tools [23]), but directly on samples coming from the distribution. This presents advantages in terms of a more flexible update of the controller, and the possibility to consider cases where the system is modelled by distributions with unbounded support.

Notation: The Kronecker product and the vectorization operation are denoted by $\otimes$ and $vec(\cdot)$. The identity matrix of size a is written as I_a, and $1_{a,b}$ $(0_{a,b})$ refers to an $a \times b$ matrix of all ones (zeros). Gaussian and uniform distributions will be indicated by $\mathcal{N}$ and $\mathcal{U}$, respectively. A Gamma distribution with mean a/b will be written as $\Gamma(a, b)$. The shorthand notation $\|v\|_A^2 = v^\top A v$ is used. Signal sequences are given without indices, e.g., $x = [x_0^\top \, x_1^\top \, \dots x_{T-1}^\top]^\top$ is a state sequence of length T. Finally, $\chi_{\mathcal{E}}$ denotes the indicator function that equals 1 if $\mathcal{E}$ holds, and 0 otherwise.

II. PROBLEM STATEMENT

The dynamic system under study is linear, time-invariant, and described by the stochastic difference equation

$$x_{t+1} = Fx_t + Gu_t + v_t, \quad x_0 = \bar{x}, \tag{1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the state and the input at time t, respectively. Matrix G is assumed known[1], while matrix F is deterministic but unknown. The process noise $\{v_t\}_t$ is a stochastic process of independent and identically distributed random vectors; their distribution $\mathbb{P}_v$ is assumed to admit a probability density $p_v(\cdot)$, which has a known analytic expression but depends on an unknown vector θ_v.

The aim is to find the state-feedback policies $\{K_t\}_{t=0}^{T-1}$ solving the Linear Quadratic Regulator (LQR) problem

$$\min_{\{K_t\}_{t=0}^{T-1}} \mathbb{E}_v \left[x_T^\top S x_T + \sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t \right] \tag{2}$$

dealing with the fact that matrix F ruling the dynamics (1) is unknown. In Section III we propose an iterative procedure that is articulated in the following three steps:

- (S-A): Bayesian identification of the system matrix F, and extraction of a credible region Θ_ϖ with probability level ϖ from its posterior distribution. We consider the one that is symmetric around the mean: this choice is arbitrary, but customary.
- (S-B): Extraction of N independent samples from Θ_ϖ, and solution of a scenario-based program returning the robust open-loop solution of the LQR problem.
- (S-C): Computation of the optimal LQR solution as state-feedback for the worst-case scenario observed in (S-B). Such control policy is then applied to the system and a new state trajectory is obtained.

The three aforementioned steps are then combined in an iterative scheme presented in Section IV. The goal is to update the control design as more knowledge of the system is gathered from its trajectories, while preserving a desired level of robustness specified by ϖ.

III. METHOD: FUNDAMENTAL STEPS

A. Bayesian model estimate (S-A)

The aim of this step is to compute the posterior distribution of the unknown state matrix F given observed state trajectories. Such a distribution will be then used to compute the posterior region Θ_ϖ.

First of all, let us rewrite model (1) in terms of $f = vec(F)$. Assume the observation of N_S state trajectories, where the $i-$th trajectory has length $N_T(i)$. Then, for $i = 1, ..., N_S$ and $t = 1, ..., N_T(i)$, model (1) can be written as

$$\begin{aligned} x_{t+1}^{(i)} &= Fx_t^{(i)} + Gu_t^{(i)} + v_t^{(i)}, \\ &= \underbrace{(x_t^{(i)} \otimes I_n)^\top}_{\phi_t^{(i)}} f_i + Gu_t^{(i)} + v_t^{(i)}. \end{aligned} \tag{3}$$

[1]This choice has been done for the sake of clarity: the overall procedure can be seamlessly extended to the case with unknown G.

For a generic trajectory i, one can write

$$\underbrace{\begin{bmatrix} x_1^{(i)} - Gu_0^{(i)} \\ \vdots \\ x_{N_T}^{(i)} - Gu_{N_T-1}^{(i)} \end{bmatrix}}_{Y^{(i)}} = \underbrace{\begin{bmatrix} \phi_0^{(i)} \\ \vdots \\ \phi_{N_T-1}^{(i)} \end{bmatrix}}_{\Phi^{(i)}} f + \underbrace{\begin{bmatrix} v_0^{(i)} \\ \vdots \\ v_{N_T-1}^{(i)} \end{bmatrix}}_{v^{(i)}}. \tag{4}$$

The overall model that combines data from all N_S trajectories is $Y^{(1:N_S)} = \Phi^{(1:N_S)} f + v^{(1:N_S)}$, where $Y^{(1:N_S)}$, $\Phi^{(1:N_S)}$ and $v^{(1:N_S)}$ are augmented vectors obtained by stacking row-wise $Y^{(i)}$, $\Phi^{(i)}$ and $v^{(i)}$, respectively, resulting in $n(\sum_{i=1}^{N_S} N_T(i)) := \bar{N}$ equations. In the remainder of the section, we will drop the superscript $(1 : N_S)$ for ease of notation.

We now want to compute the posterior $f|Y$, denoted by $\mathbb{P}_f$, from which we will extract the credible region Θ_ϖ such that

$$\mathbb{P}_f(f \in \Theta_\varpi) = \varpi$$

and is symmetric around the mean.
To compute the posterior, we have to specify the likelihood and the prior. Given the value of f, data follow the process noise distribution $\mathbb{P}_v$: hence, the likelihood is $p_v(Y|f)$. As regards the prior $\mathbb{P}_f^{(0)}$, we assume that such a distribution admits a density $p_f(\cdot)$ whose expression is known but depends on a parameter θ_f that is fixed and unknown. The dependence among f, θ_f, θ_v and Y is presented in the Bayesian network in Figure 1.

Remark 1: Note that the presented framework is very general, because it can accommodate any (possibly multi-modal) distribution. However, typical choices involve uni-modal distributions such as the Gaussian and the Laplacian. In these cases, θ_f and θ_v may contain the unknown mean and covariance of f and each $v_t^{(i)}$, respectively. For example, Laplacian noises are typically used to robustly handle outliers [24], [25], [26], while f is modelled as Laplacian when sparsity needs to be promoted [27], [28], [29]. □

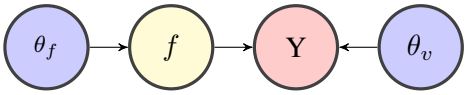

Fig. 1. Bayesian network associated to step (S-A).

According to these hypotheses, one can derive an analytic formula for the posterior density function of $f|Y$ depending on the parameters θ_f and θ_v: in fact, the main ingredient is the product of the likelihood $p_v(Y|f;\theta_v)$ and the prior $p_f(f;\theta_f)$, whose expressions are known. At this point, two difficulties arise:

- parameters θ_f and θ_v are unknown;
- even if θ_f and θ_v were known, retrieving Θ_ϖ may be far from trivial because it involves an integral which is difficult to evaluate if the posterior distribution has a complicated expression.

Both these problems will be effectively tackled by resorting to Markov Chain Monte Carlo (MCMC) [30].

To address the first issue, we adopt the Empirical Bayes paradigm [31] and estimate parameters θ_f and θ_v from data by optimizing the marginal likelihood: taking its negative logarithm, the estimate is obtained by solving

$$\min_{\theta_f,\theta_v} -\log p_v(Y|\theta_f,\theta_v). \tag{5}$$

However, problem (5) is typically nonconvex and potentially high dimensional: thus, deterministic optimization routines, due to their sensitivity to initial conditions, can be inaccurate. This issue can be overcome by deploying an MCMC scheme acting as follows.
Note first that (dropping subscripts for notational ease)

$$\begin{aligned} p(f,\theta_f,\theta_v|Y) &\propto p(f|\theta_f,\theta_v,Y)p(\theta_f,\theta_v|Y), \\ &\propto p(f|\theta_f,\theta_v,Y)p(Y|\theta_f,\theta_v). \end{aligned} \tag{6}$$

That is, the marginal likelihood can be explored by drawing samples from $p(f,\theta_f,\theta_v|Y) := \pi(f,\theta_f,\theta_v)$. The rationale of MCMC is then to build a Markov chain whose invariant distribution is $\pi(f,\theta_f,\theta_v)$. This effectively provides a strategy to draw samples from this target distribution that are principled candidates to solve problem (5). To this aim, one procedure that can be adopted is known as Metropolis-Hastings [30], which is summarized in Algorithm 1.

Algorithm 1 Metropolis-Hastings to explore $\pi(\theta)$, with $\theta = (f,\theta_f,\theta_v)$. Input: proposal density $q(\cdot|\cdot)$, target density $\pi(\cdot)$. Output: samples $\{\theta_j\}$ from $\pi(\cdot)$.

Initialize θ_0 and set $j=0$;
while not converged **do**
  sample a point $\bar\theta$ from the proposal density $q(\cdot|\theta_j)$;
  sample a variable $u \sim \mathcal{U}([0,1])$;
  compute the acceptance probability

$$\mathbf{p}_a = \min\left(1, \frac{\pi(\bar\theta)q(\theta_j|\bar\theta)}{\pi(\theta_j)q(\bar\theta|\theta_j)}\right)$$

  if $u \le \mathbf{p}_a$ **then**
    set $\theta_{j+1} = \bar\theta$;
  else
    set $\theta_{j+1} = \theta_j$;
  end if
  $j \leftarrow j+1$.
end while

Updating together the variables f, θ_v, and θ_f is often difficult, because finding an effective proposal distribution $q(\cdot|\cdot)$ (see Algorithm 1) can be far from trivial. The proposed strategy, introduced in [32], is known as the *single-component Metropolis-Hasting* algorithm [30]. Instead of sampling $\pi(f,\theta_f,\theta_v)$, the conditional posterior distributions (also known as full-conditionals) are sequentially sampled. Using the Bayesian network of Figure 1, these are

$$\begin{aligned} &\bullet\ \pi(f|\theta_f,\theta_v) \propto p(Y|f,\theta_v,\theta_f)p(f|\theta_f), \\ &\bullet\ \pi(\theta_v|\theta_f,f) \propto p(Y|f,\theta_f,\theta_v)p(\theta_v), \\ &\bullet\ \pi(\theta_f|\theta_v,f) \propto p(f|\theta_f)p(\theta_f). \end{aligned} \tag{7}$$

Specifically, in our setup the single-component Metropolis-Hastings algorithm proceeds as follows: assuming starting from values $(\bar f,\bar\theta_f,\bar\theta_v)$; then, one draws $\tilde f$ from $\pi(f|\bar\theta_v,\bar\theta_f)$, $\tilde\theta_f$ from $\pi(\theta_f|\tilde f,\bar\theta_v)$, $\tilde\theta_v$ from $\pi(\theta_v|\tilde\theta_f,\tilde f)$, and then iterates again starting from $(\tilde f,\tilde\theta_v,\tilde\theta_f)$. Since all prior distributions admit a density, the full-conditionals are well-defined: hence, the procedure is guaranteed to generate the Markov chain with invariant distribution $\pi(f,\theta_f,\theta_v)$ after some burn-in iterates.
It can happen that the distributions on the right-hand sides of (7) are conjugate priors [33], yielding a full-conditional that can be sampled without resorting to Algorithm 1. If this is the case for all full-conditionals, then the single-component Metropolis-Hastings goes under the name of *Gibbs sampler* [34]. The details for a particular case that will be useful in the remainder of the paper are provided.

Example 1: Consider the distributions $f \sim \mathcal{N}(0,\lambda P_f)$ and $v \sim \mathcal{N}(0,\sigma^2 P_v)$ for each $t=1,...,N_T(i)$ and $i=1,...,N_S$. We assume that $P_f \in \mathbb{R}^{n^2\times n^2}$ and $P_v \in \mathbb{R}^{\bar N\times\bar N}$ are known, thus in this case it holds that $\theta_f=\lambda$ and $\theta_v=\sigma^2$. We give to precisions λ^{-1} and σ^{-2} a proper Gamma prior approximating an uninformative distribution over the positive real axis.
Consider then the full conditionals and their expressions in terms of likelihood-prior products presented in (7). As for f, note that both prior $p(f|\lambda)$ and likelihood $p(Y|f,\sigma^2)$ are Gaussian. It can be verified that this distribution is self-conjugate, and one obtains

$$\begin{aligned} &(f|\lambda,\sigma^2,Y) \sim \mathcal{N}(\hat\mu,\hat\Sigma), \quad \text{where} \\ &\begin{cases} \hat\mu = (\Phi^\top P_v^{-1}\Phi/\sigma^2 + P_f^{-1}/\lambda)^{-1}\Phi^\top P_v^{-1}Y/\sigma^2, \\ \hat\Sigma = (\Phi^\top P_v^{-1}\Phi/\sigma^2 + P_f^{-1}/\lambda)^{-1}. \end{cases} \end{aligned} \tag{8}$$

The full conditionals for λ^{-1} and σ^{-2} also admit a simple representation: indeed, they involve a Gaussian likelihood and a Gamma prior which are known to be a conjugate pair. Specifically, one obtains

$$p(\lambda^{-1}|f) \sim \Gamma\left(\frac{n^2}{2}, \frac{\|f\|^2_{P_f^{-1}}}{2}\right), \tag{9}$$

$$p(\sigma^{-2}|Y,f) \sim \Gamma\left(\frac{\bar N}{2}, \frac{\|Y-\Phi f\|^2_{P_v^{-1}}}{2}\right). \tag{10}$$

The Gibbs sampler for this case is then summarized in Algorithm 2.

Algorithm 2 Gibbs sampler to explore $\pi(f, \lambda, \sigma^2)$ in Example 1. Input: number of iterations N_g, data Y and Φ, full conditional densities. Output: samples from $\pi(f, \lambda, \sigma^2)$.

for $i = 1, ..., N_g$ **do**
  if $i = 1$ **then**
    set $f(i) = (\Phi^\top \Phi)^{-1} \Phi^\top Y$;
  else
    sample $f(i)$ from (8);
  end if
  sample λ^{-1} from (9);
  sample σ^{-2} from (10);
end for

□

Up to this point, a procedure to sample the posterior $p(f, \theta_f, \theta_v|Y)$ has been presented. Now, the marginal likelihood optimization problem of (5) is addressed in the following way: we obtain N_c samples from the posterior by running the (single-component) Metropolis-Hastings algorithm, and use them to optimize the marginal likelihood (5). The selected optimal hyperparameters can be then plugged into the full conditional of f to obtain the posterior $p(f|Y, \theta_f, \theta_v)$. Next, the estimated parameters θ_f and θ_v fully specify the posterior $f|Y$, and our goal is to extract from such distribution a credible region with probability level ϖ. The proposed approach is to leverage again the MCMC scheme to draw samples from the posterior $\mathbb{P}_f$ and obtain in sampled form the $[1 - \varpi/2, \varpi/2]$-quantiles.

B. Robust optimization of open-loop sequences (S-B)

Having obtained in the previous step a probabilistic estimate of f, we propose to solve problem (2) via the following robust convex program:

$$\begin{aligned} \min_{u,x} \ \max_{\tilde{f} \in \Theta_\varpi} \ & \mathbb{E}_v[x_T^\top S x_T + \sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t], \\ \text{s.t.} \quad & x_{t+1} = \tilde{F} x_t + G u_t + v_t, \quad x_0 = \bar{x}, \\ & \tilde{f} = vec(\tilde{F}) \in \Theta_\varpi, \end{aligned} \tag{11}$$

That is, we want to find a solution of the LQR problem that is robust with respect to the posterior measure $\mathbb{P}_f$. First of all, since the optimal state-feedback solving (11) does not depend on the process noise $\{v_t\}_t$ [35], we can consider

$$\min_{u,x} \ \max_{\tilde{f} \in \Theta_\varpi} \ x_T^\top S x_T + \sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t, \tag{12a}$$

$$\text{s.t.} \quad x_{t+1} = \tilde{F} x_t + G u_t, \quad x_0 = \bar{x}, \tag{12b}$$
$$\tilde{f} = vec(\tilde{F}) \in \Theta_\varpi.$$

We can further manipulate (12) by eliminating the dependence on the state sequence. In fact, the dynamics (12b) yield the state evolution

$$x_t = \tilde{F}^t \bar{x} + \sum_{s=0}^{t-1} \tilde{F}^{t-1-s} G u_s \tag{13}$$

which, substituted into (12a), yields the unconstrained program

$$\arg\min_u \ \max_{\tilde{f} \in \Theta_\varpi} \ u^\top B(\tilde{f}) u + 2a^\top(\tilde{f}) u, \tag{14}$$

where $B(\tilde{f}) \in \mathbb{R}^{mT \times mT}$ and $a(\tilde{f}) \in \mathbb{R}^{mT \times 1}$ are defined block-wise, with T blocks that are, for $i, j = 1, \ldots, T$ (dropping the dependence on $\tilde{f}$ for notational ease),

$$\begin{aligned} \left(a^\top\right)_i &= \sum_{t=0}^{T-1} \bar{x}^\top (\tilde{F}^t)^\top Q \tilde{F}^{t-i} G \chi_{t-i \geq 0} + \bar{x}^\top (\tilde{F}^T) S \tilde{F}^{T-i} G, \\ (B)_{i,j} &= \sum_{t=0}^{T-1} G^\top (\tilde{F}^{t-i})^\top Q \tilde{F}^{t-j} G \chi_{t-i \geq 0 \wedge t-j \geq 0} \\ &\quad + G^\top (\tilde{F}^{T-i})^\top S \tilde{F}^{T-j} G + R \chi_{i=j}. \end{aligned}$$

At this point, (14) can be rewritten in epigraph form as

$$\begin{aligned} &\arg\min_{\tau, u} \ \tau \\ &\text{s.t. } u^\top B(\tilde{f}) u + 2a^\top(\tilde{f}) u \leq \tau \\ &\text{for all } \tilde{f} \in \Theta_\varpi. \end{aligned} \tag{15}$$

Now, since Θ_ϖ is (usually) a compact subset of $\mathbb{R}^{n^2}$, program (15) involves an infinite number of constraints and is therefore hard to solve. For this reason, we resort to a sample-based relaxation leveraging the scenario approach [21], [36] and consider

$$\begin{aligned} &\arg\min_{\tau, u} \ \tau \\ &\text{s.t. } u^\top B(f^{(k)}) u + 2a^\top(f^{(k)}) u \leq \tau \\ &\text{for all } k = 1, \ldots, N. \end{aligned} \tag{16}$$

The samples from the posterior $\{f^{(k)}\}_{k=1}^N$ are drawn by running in parallel the MCMC scheme proposed in step (S-A) with different initializations, and discarding the ones that do not belong to Θ_ϖ. With this construction, the N samples are independent following the conditional probability measure defined as

$$\mathbb{P}'_f(A) = \frac{\mathbb{P}_f(A \cap \Theta_\varpi)}{\mathbb{P}_f(\Theta_\varpi)} = \frac{\mathbb{P}_f(A \cap \Theta_\varpi)}{\varpi} \tag{17}$$

for any event A in the $\sigma-$field under study. Now, the scenario approach theory comes into play to characterize the generalization capability of the optimal solution $\hat{u}$ of (16) based on N samples. In particular, one is interested in bounding for the violation probability

$$\mathbb{P}'_f\left(\tilde{f} : \hat{u}^\top B(\tilde{f}) \hat{u} + 2a^\top(\tilde{f}) \hat{u} > \hat{\tau}\right), \tag{18}$$

which is the probability to observe a new realization of the uncertainty that violates the constraint evaluated at the current solution $(\hat{u}, \hat{\tau})$. The main result is stated as follows: if samples $\{f^{(k)}\}_{k=1}^N$ are independent and identically distributed, and

$$N \geq \frac{2}{\epsilon}\Big(\ln \frac{1}{\beta} + mT\Big), \tag{19}$$

then the violation probability is smaller than ϵ with confidence $1 - \beta$ [36].

C. Computation of the robust state-feedback solution (S-C)

The solution $\hat{u}$ to (16) obtained in step (S-B) can be interpreted as the open-loop solution of the LQR problem in the worst-case scenario within the samples drawn from Θ_{ϖ}. To allow for possible disturbance rejection, we seek a closed-loop solution. We obtain this by considering the state matrix yielding the worst-case scenario in (16), denoting it F_{wc}, and computing the classic LQR solution by means of the Riccati difference equation [11], [37], [38]. Specifically, the latter consists in finding a matrix sequence $\{M_t\}_{t=0}^{T}$ obtained through the following backward recursion:

$$\begin{cases} M_T = S \\ Mt = Q + F_{wc}^{\top} M_{t+1} F_{wc} \\ \qquad -F_{wc}^{\top} M_{t+1} G(R + G^{\top} M_{t+1} G)^{-1} G^{\top} M_{t+1} F_{wc}. \end{cases}$$

Next, the optimal feedback policy is obtained as

$$K_t = -(R + G^{\top} M_{t+1} G)^{-1} G^{\top} M_{t+1} F_{wc}. \qquad (20)$$

IV. METHOD: UPDATING SCHEME

In this Section, steps (S-A), (S-B) and (S-C) detailed in Section III are combined in an updating scheme. After having collected a first state trajectory, one applies the three steps sequentially; the idea is then to deploy the sequence of feedback gains computed in step (S-C) to obtain a new system trajectory, and repeat the procedure. By adding new data to the existing set, the system's estimate (S-A) can be refined, and by doing so samples that are closer to the true, unknown system can be used in the design procedure (S-B). In fact, in the spirit of the Bernstein-von Mises' Theorem and general consistency results for Bayesian estimates (see, e.g., Chapter 10 in [39]), the posterior $f|Y$ converges to the point distribution centred at the true value f: from this it follows that the size of the posterior regions $\{\Theta_{\varpi}^{(i)}\}_i$ is going to decrease (with respect to the usual Lebesgue measure) as the number of iterations i increases, while keeping the same probability level ϖ. Importantly, the scenario program in (S-B) is solved keeping N, ϵ and β constant (see (19)) at each iteration: thus, the same guarantee on the robustness holds, but conservatism is reduced because dispersion in the samples is decreasing as the number of updates increases.

We now examine in more detail the proposed iterative application of steps (S-A), (S-B) and (S-C): the aim is to find a feedback policy $\{K_t\}_{t=0}^{T-1}$ solving the original problem (2), being robust against the uncertainty deriving from estimating the unknown system matrix.

The user defines the number of times N_{it} the procedure is to be repeated. At each iteration, a sequence of feedback matrices $\{K_t^{(i)}\}_{t=0}^{T-1}$ solving (16) in (S-B) is obtained. To compare the performance throughout the iterations (e.g., by means of the cost values), it is key to solve (S-B) considering the same initial condition $x_0 = \bar{x}$. This is also reasonable because one is typically interested in the behaviour at a certain fixed working condition of the plant. At the end of the N_{it} iterations, there can be two possible decision rules to select the feedback policy $\{\hat{K}_t\}_{t=0}^{T-1}$ among the available N_{it}: either one selects the one obtained for iteration $k = N_{it}$, or one chooses the one corresponding to the iteration yielding the smallest minimax cost of (16). Typically the two solutions coincide if N_{it} is sufficiently large; however, the second option turns out to be the most reliable in order not to be sensitive to the choice of N_{it}.

The overall update-and-design procedure is summarized in Algorithm 3.

Algorithm 3 Update-and-design scheme for robust LQR. Input: system matrix G, initial state $\bar{x}$, state trajectory length N_T, probability level ϖ, LQR cost matrices Q, R, S and horizon T, number of samples N, parameter μ, number of iterations N_{it}. Output: Feedback policy $\{\hat{K}_t\}_{t=0}^{T-1}$.

for $i = 1, \ldots, N_{it}$ **do**
  if $i = 1$ **then**
    Set $u_t^{(i)} \sim \mathcal{N}(0, I_m)$, $t = 0, \ldots, T-1$;
  else
    Set $u_t^{(i)} = K_t^{(i-1)} x_t^{(i)}$, $t = 0, \ldots, T-1$;
  end if
  Compute data matrices $Y^{(1:i)}$, $\Phi^{(1:i)}$;
  Step (S-A): retrieve the posterior $\mathbb{P}_f^{(i)}$ for f;
  Retrieve $\Theta_{\varpi}^{(i)}$ (in sampled form);
  Draw N independent samples from $\mathbb{P}_f'^{(i)}$ (see (17));
  Step (S-B): compute $\hat{u}^{(i)}$ solving (16);
  Identify the worst-case scenario $F_{wc}^{(i)}$ yielding $\hat{u}^{(i)}$;
  Step (S-C): compute nominal LQR solution for $F_{wc}^{(i)}$;
end for
select $\{K_t^{(i)}\}_{t=0}^{T-1}$ yielding the lowest minimax cost (16);

Remark 2: To generate new state trajectories, several choices can be made for the input sequence. Three possible strategies are:

- the one we propose, i.e., applying the feedback sequence computed at the previous iteration on the system at its current state;
- exciting the system via white noises;
- using pre-stabilizing feedbacks.

In the latter two cases, the length of the state trajectory N_T used for step (S-A) is independent of the LQR horizon T. On the other hand, updating the system estimate via the feedback computed at step (S-C) would apparently require N_T to be equal to T. This can be avoided as follows: for $N_T < T$, by considering only the first N_T feedback matrices; and for $N_T > T$, by using the last feedback matrix K_{T-1} for all future times $t = T, \ldots, N_T$. The first two strategies may be preferred to enhance exploration, but may incur stability issues for "large" values of N_T; a thorough discussion on this aspect is however beyond of the scope of this paper. □

V. NUMERICAL EXPERIMENTS

This section shows the application of the proposed update-and-design scheme to randomly sampled plants of different sizes. The goal is on the one hand to demonstrate the effectiveness of the algorithm and on the other to provide insights by showing some of its peculiar features.

In all the following tests, the true vectorized state matrix F will be sampled from a multivariate Gaussian distribution with zero mean and covariance $0.5I_{n^2}$. The process noise is an i.i.d. sequence where each sample v_t is drawn from $\mathcal{N}(0, 0.3I_n)$ for all t. Under these assumptions, step (S-A) coincides with the Gibbs sampler presented in Example 1. Finally, the matrix G is equal to $1_{n,m}$ and the initial condition is $\bar{x} = 0.5 \cdot 1_{n,1}$. All state trajectories have the same length N_T, which is taken to be equal to the LQR horizon T. The LQR cost matrices are $Q = 10I_n$, $S = 8I_n$ and $R = I_m$. As for the scenario approach robustness guarantees, we fix $\epsilon = 0.1$ and $\beta = 0.1$: this means that with 90% confidence the solution of step (S-B) has a violation probability of 10%. We compute the corresponding value of N from (19) and draw samples from the posterior region computed at step (S-A). Precisely, we consider the $\varpi = 0.977$ probability region to be symmetric around the posterior mean. The procedure summarized in Algorithm 3 is initialized by using an open-loop state trajectory of length N_T obtained with white Gaussian noise, and is repeated $N_{it} = 15$ times using as inputs the feedback policies computed at the previous iterations, but applied on the current state.

When assessing the performance of the computed feedback policy, there are three quantities involved. Denoting with $J(\bar{x}, \{u_t\}_{t=0}^{T-1}; F) = x_T^\top S x_T + \sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t$, where $x_{t+1} = Fx_t + Gu_t$, these are the following:

- $\bar{\mathcal{L}} = J(\bar{x}, \{u_t^*\}_{t=0}^{T-1}; F)$, i.e., the optimal LQR cost under the true, noiseless, dynamics.
- $\mathcal{L}^{(i)} = J(\bar{x}, \{K_t^{(i)} x_t^{(i)}\}_{t=0}^{T-1}; F)$, i.e., the suboptimal LQR cost obtained at iteration i by using the estimated feedback policy on the true system;
- $\hat{\mathcal{L}} = J(\bar{x}, \{\hat{K}_t \hat{x}_t^{(i)}\}_{t=0}^{T-1}; F)$, i.e., the value returned following the decision rule proposed in Algorithm 3. That is, $\hat{\mathcal{L}} = \min_{i=1,\dots,N_{it}} \mathcal{L}^{(i)}$.

Note that these scores are not observable, because they depend on the unknown state matrix F: however, we use them to study the performance of the proposed approach.

A. Sample performance on a single plant

The goal of this test is to show how two features of Algorithm 3 evolve across the iterations: the system's estimation performed in step (S-A), and the estimated LQR cost $\mathcal{L}^{(i)}$ associated with the controller designed at step (S-C).

In this experiment, state and input dimensions are set to $n = 4$ and $m = 1$, respectively, and the true state-matrix is

$$F = \begin{bmatrix} 0.528 & 0.561 & 0.996 & -0.790 \\ -0.375 & 0.599 & 0.827 & 0.700 \\ -0.977 & 0.953 & -0.315 & 0.300 \\ 0.827 & 0.764 & 0.558 & 0.667 \end{bmatrix}.$$

The state trajectory length N_T is 5.

Figure 2 depicts the posterior distribution information of f obtained via step (S-A) at the first and last iteration of the algorithm. It can be noted that uncertainty around the estimates shrinks thanks to new data. Thus, since the same number N of samples is drawn from uncertainty regions at each iteration, conservatism is effectively reduced. This is apparent in Figure 3, where the true LQR cost $\bar{\mathcal{L}}$ is compared with $\mathcal{L}^{(i)}$. It can be noted that the suboptimal LQR costs tend to decrease and reach the optimal LQR cost at the end of the procedure. Moreover, it can be noticed that the proposed decision rule for selecting the optimal estimated cost is effectively capable of selecting the minimum one, which is 1.8% higher with respect to the true LQR score.

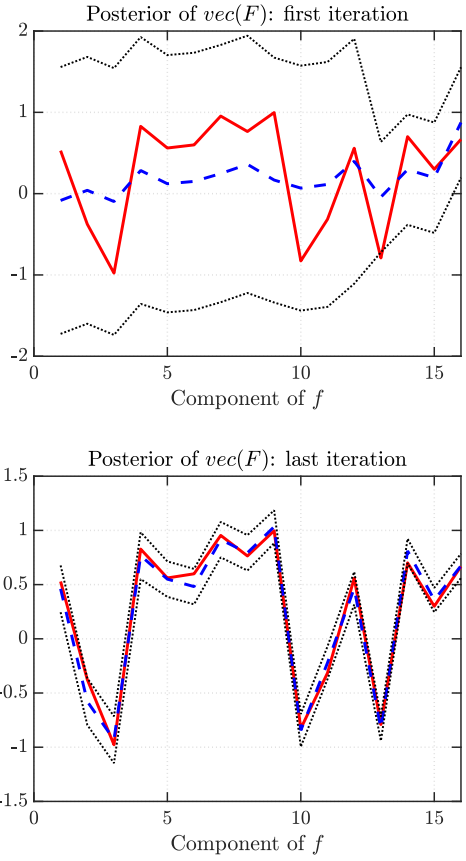

Fig. 2. Estimation performance of $f = vec(F)$ at the first (top panel) and at the last (bottom panel) iteration of Algorithm 3. LEGEND: Solid red: true f; Dashed blue: estimated f (posterior mean); Dotted black: symmetric posterior region bounds with probability level $\varpi = 0.997$.

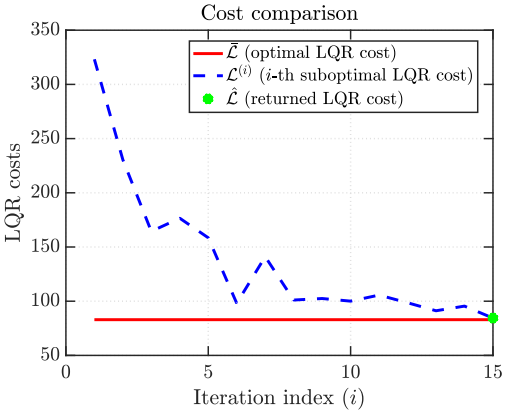

Fig. 3. Evolution of estimated LQR costs $\mathcal{L}^{(i)}$ throughout Algorithm 3.

B. Monte Carlo study

The performance of Algorithm 3 is now studied on 50 random systems where $vec(F)$ is sampled from $\mathcal{N}(0, 0.5I_{n^2})$. In this test, $n = 2$ and $m = 1$ and $N_T = 5$.
First, the algorithm's capability of improving on the LQR cost estimated at the first iteration is analyzed. This can be interpreted as a measure of the decrease in conservatism offered by the proposed algorithm with respect to data-agnostic robust LQR design schemes proposed in [22]. Recalling the definitions presented at the beginning of Section V, we consider the following performance metric:

$$\mathcal{M} = 100\% \frac{\mathcal{L}^{(1)} - \hat{\mathcal{L}}}{\hat{\mathcal{L}}}.$$

We study the statistics of this metric in Figure 4. Evaluated at each abscissa Z, the plot returns the number of runs such that the relative improvement $\mathcal{M}$ is greater than or equal to $Z\%$. There are 3 runs that yield a negative value for $\mathcal{M}$, but 11 that exceed the 100% improvement. The $\mathcal{M}$-values of the first are $[-2.36, -1.02, -0.45]$, and the quartiles of the latter are $[202, 460, 624]$. Focusing on the index in $\{1, \dots, N_{it}\}$ that the algorithm selected as the optimal one (i.e., returning the lowest minimax cost), its mean value over all 50 runs is 11, and its quartiles are $[7, 11, 14]$.

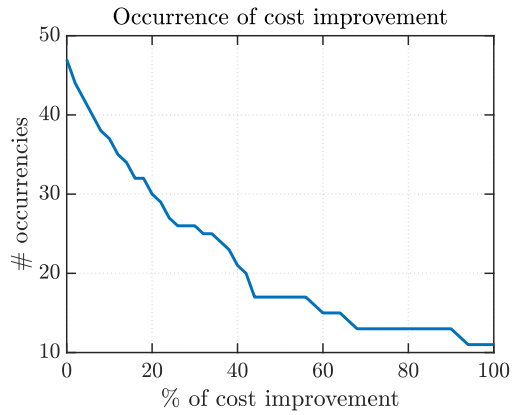

Fig. 4. Number of occurrencies (over all 50 runs of the Monte Carlo experiment) of event $\{\mathcal{M} \geq Z\%\}$ for each abscissa $Z \in [0, 100]$.

Next, we investigate the conservatism of the controller obtained from Algorithm 3 by comparing $\hat{\mathcal{L}}$ with the optimal cost $\bar{\mathcal{L}}$ of the corresponding plant. The results are displayed in Figure 5. It can be seen that the costs of the proposed feedback laws $\hat{\mathcal{L}}$ are very close to the optimal ones, showcasing the expected behaviour in terms of conservatism reduction. To better view the statistics of such a performance, Figure 6 presents the boxplot of the relative difference (i.e., normalized with respect to the optimal cost $\bar{\mathcal{L}}$) between the two costs.

VI. CONCLUSIONS AND FORTHCOMING RESEARCH

In the context of the LQR control of unknown linear systems, we propose a three-step algorithm to compute a time-varying state-feedback control law that guarantees a robust

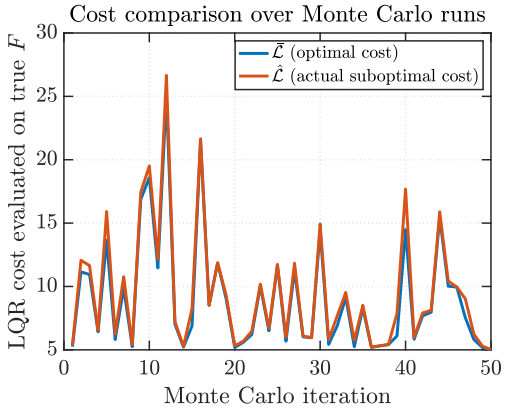

Fig. 5. Comparison of LQR scores $\bar{\mathcal{L}}$ and $\hat{\mathcal{L}}$.

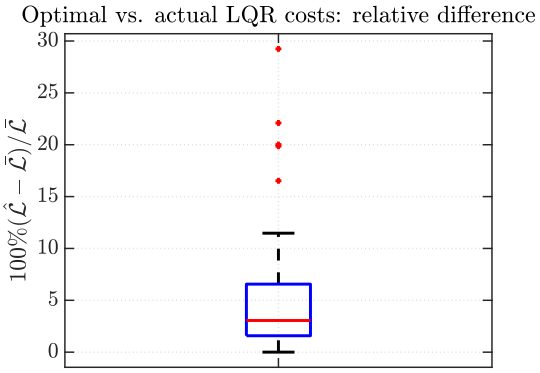

Fig. 6. Relative difference between $\bar{\mathcal{L}}$ and $\hat{\mathcal{L}}$.

performance and is updated by exploiting the availability of new data. Results show that the proposed procedure, by leveraging its refined knowledge of the true system, shall outperform data-agnostic methods by reducing conservatism in robust design.

Future analyses involve tests with non-Gaussian distributions, building upon the theory presented in Section III-A, and a more thorough convergence analysis of the MCMC schemes. Also, the dependence of the overall performance from the LQR horizon length should be studied.
The proposed procedure could be extended to the case in which state measurements are available only through noisy outputs, or some of them are missing. Moreover, since the current formulation can also take input/state constraints into account, another intriguing direction involves the adaptation of the proposed procedure to (dual) Model Predictive Control; in this case, scalability needs to be discussed and improved.

ACKNOWLEDGMENTS

The authors would like to thank Professor Melanie Zeilinger and her research group (in particular, Johannes Köhler, Elena Arcari, Kim Wabersich and Andrea Zanelli) for the insightful discussions about this work.

REFERENCES

[1] L. Ljung, *System identification: theory for the user*. Prentice Hall, 1999.

[2] I. Markovsky, J. C. Willems, S. V. Huffel, and B. D. Moor, *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. Society for Industrial and Applied Mathematics, 2006.

[3] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. MIT Press, 1998.

[4] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[5] S. Formentin, K. van Heusden, and A. Karimi, "A comparison of model-based and data-driven controller tuning," *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 10, pp. 882–897, 2014.

[6] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 253–279, 2019.

[7] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct & indirect data-driven control formulations via regularizations and relaxations," *arXiv: 2101.01273*, 2021.

[8] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice-Hall, Inc., 1996.

[9] A. Karimi, H. Khatibi, and R. Longchamp, "Robust control of polytopic systems by convex optimization," *Automatica*, vol. 43, no. 8, pp. 1395–1402, 2007.

[10] B. Barmish, *New Tools for Robustness of Linear Systems*. Prentice Hall PTR, 1994.

[11] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1990.

[12] S. K. Jha, S. B. Roy, and S. Bhasin, "Data-driven adaptive lqr for completely unknown lti systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4156–4161, 2017.

[13] J. Fong, Y. Tan, V. Crocher, D. Oetomo, and I. Mareels, "Dual-loop iterative optimal control for the finite horizon lqr problem with unknown dynamics," *Systems & Control Letters*, vol. 111, pp. 49–57, 2018.

[14] J. Umenberger and T. B. Schön, "Learning convex bounds for linear quadratic control policy synthesis," in *NeurIPS*, 2018.

[15] G. R. Gonçalves da Silva, A. S. Bazanella, C. Lorenzini, and L. Campestrini, "Data-Driven LQR Control Design," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 180–185, 2019.

[16] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, vol. 20, Aug 2019.

[17] N. Li, I. Kolmanovsky, and A. Girard, "LQ control of unknown discrete-time linear systems—A novel approach and a comparison study," *Optimal Control Applications and Methods*, vol. 40, no. 2, pp. 265–291, 2019.

[18] B. Pang, T. Bian, and Z.-P. Jiang, "Adaptive dynamic programming for finite-horizon optimal control of linear time-varying discrete-time systems," *Control Theory and Technology*, vol. 17, pp. 73–84, 02 2019.

[19] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[20] A. Iannelli and R. S. Smith, "A Multiobjective LQR Synthesis Approach to Dual Control for Uncertain Plants," *IEEE Control Systems Letters*, vol. 4, no. 4, pp. 952–957, 2020.

[21] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

[22] A. Scampicchio, A. Aravkin, and G. Pillonetto, "Stable and Robust LQR Design via Scenario Approach," *Automatica (to appear)*, 2021.

[23] C. Scherer and S. Weiland, *Linear Matrix Inequalities in Control*. Lecture Notes, 2000.

[24] P. J. Huber, *Robust Statistics*. New York, NY, USA: John Wiley and Sons, 1981.

[25] G. A. Hewer, R. D. Martin, and J. Zeh, "Robust preprocessing for kalman filtering of glint noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 23, no. 1, pp. 120–128, 1987.

[26] A. Aravkin, B. Bell, J. Burke, and G. Pillonetto, "An ℓ_1-laplace robust Kalman smoother," *IEEE Trans. on Automatic Control*, vol. 56, no. 12, pp. 2898–2911, 2011.

[27] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society, Series B.*, vol. 58, pp. 267–288, 1996.

[28] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, pp. 1348–1360, 2001.

[29] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[30] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in Practice*. London: Chapman and Hall, 1996.

[31] J. S. Maritz and T. Lwin, *Empirical Bayes Method*. Chapman and Hall, 1989.

[32] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[33] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis, Second Edition*. Taylor & Francis, 2003.

[34] A. E. Gelfand and A. F. M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398–409, 1990.

[35] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA, USA: Athena Scientific, 2005, vol. I.

[36] G. C. Calafiore, "Random convex programs," *SIAM Journal on Optimization*, vol. 20, pp. 3427–3464, 09 2010.

[37] R. Bitmead and M. Gevers, "Riccati difference and differential equations: Convergence, monotonicity and stability," in *The Riccati Equation*, S. Bittanti, A. Laub, and J. Willems, Eds., 1991.

[38] A. E. Bryson and Y. Ho, *Applied optimal control: optimization, estimation, and control*. Hemisphere Publishing Corporation, 1975.

[39] A. W. van der Vaart, *Asymptotic Statistics*. Cambridge University Press, 1998.