Sensitivity and safety of fully probabilistic control

Bernat Guillen Pegueroles$^{1,*}$ and Giovanni Russo$^{2,*}$

Abstract—In this paper we present a sensitivity analysis for the so-called fully probabilistic control scheme. This scheme attempts to control a system modeled via a probability density function (pdf) and does so by computing a probabilistic control policy that is optimal in the Kullback-Leibler sense. Situations where a system of interest is modeled via a pdf naturally arise in the context of neural networks, reinforcement learning and data-driven iterative control. After presenting the sensitivity analysis, we focus on characterizing the convergence region of the closed loop system and introduce a safety analysis for the scheme. The results are illustrated via simulations.

This is the preliminary version of the paper entitled “On robust stability of fully probabilistic control with respect to data-driven model uncertainties” that will be presented at the 2019 European Control Conference.

I. INTRODUCTION

Over the past few years, much research attention has been devoted to the design of model-free and iterative control algorithms that are able to learn how to control a given system of interest by learning from past iterations, see e.g. [1] and references therein for a recent application of iterative learning to Model Predictive Control. At the same time, for many cyber-physical systems, driven by the recent explosion in the amount of available data (via e.g. the Internet of Things) and by the dramatic improvements in computational and communication infrastructures, Deep Learning techniques, and in particular neural networks, have been increasingly used to model and classify systems [2]. In practice, the output of a deep neural network is often a probability density function (pdf), describing the state of a given system.

In this context, designing control algorithms for uncertain systems and subject to certain safety and convergence constraints is rapidly becoming a major research topic. This is motivated by a number of applications, including reinforcement learning [3] and model-free control schemes subject to state constraints [4] and with convergence guarantees, see e.g. [5] for an overview.

The design of control strategies in the absence of reliable models and in the presence of strong uncertainty has long been the subject of stochastic control, see e.g. [6]. Stochastic control is deeply related to decision science and, in particular, to Bayesian dynamic decision making, see e.g. [7], where the control action is computed by minimizing the expected value of a loss function embedding the control goal. The fully probabilistic control algorithm, which was originally introduced in the seminal work [8], belongs to the family of stochastic control algorithms, with the main difference that it selects randomized control laws that make the entire joint distribution of closed-loop variables as close as possible (in the sense of the Kullback-Leibler divergence) to their desired distribution. See also [9], [10], [11] for recent developments on this topic, together with [12], [13] for the development of an adaptive critic fully probabilistic control, which is closely related to reinforcement learning schemes based on actor critic [14].

In this paper we present a sensitivity and safety analysis for the fully probabilistic control, when the pdfs of the closed loop system are all multivariate Gaussians. In particular, we analyze how error models can propagate through the system and provide a method, based on numerical continuation, to compute the corresponding region of convergence. The region of convergence is a region of the parameter space in which the stability of the closed loop system is guaranteed via the fully probabilistic control. Then, we move onto defining a notion of safety for the system. This notion is related to the maximum error between the target state and the current state of the system. Given this definition and a safety requirement on the closed loop system, we show how it is possible to numerically compute a safety region. This is a region of the parameter space where the norm of the state never violates the safety requirement. Finally, we show how embedding learning mechanisms in the closed loop system can indeed extend both the convergence and the safety regions. The results are illustrated via an example. The complete proofs of the technical results, extended to a non-Gaussian setting, will be presented elsewhere. Also, we do not discuss here the computational aspects of the approach presented in this paper. This aspect will be discussed elsewhere.

II. PROBLEM SET-UP

Notation and initial remarks

The notation used in this paper is closely related to the one of [11]. Let $S^*$ be a given finite set, we denote by $\hat{S}$ its cardinality. The value of a given quantity, say $q$, at time $t$ is denoted by $q_t$ and the set $t^* := \{1, \ldots, \hat{t}\}$ is a given time horizon. Recall that, given the probability space $(\Omega, \Sigma, P)$ (where $\Omega$ is the sample space, $\Sigma$ is the collection of all the events, and $P$ is the probability measure) a random variable is a measurable function $X : \Omega \to \mathbb{R}$ and we denote by $E[X]$ the expected value of $X$. Also, we denote by $f(\cdot | \cdot)$ a given conditional probability density function (pdf). Then, in the context of this paper, a system is specified in the probabilistic sense, i.e. the time evolution of the system is specified via the pdf $f(x_t | u_t, x_{t-1})$, where: (i) $x_t$ is the $n$-dimensional

---

1 Princeton University, Princeton bernatp@princeton.edu
2 University College Dublin giovanni.russo1@ucd.ie
* Work done in part while the authors were with IBM Research
observed state of the system at time \( t \); (ii) \( u \) is the \( n \)-dimensional control input at time \( t \). As usual, we denote the multivariate normal distribution of the random vector \( v = [v_1, \ldots, v_n]^T \) by \( v \sim \mathcal{N} (\mu, \Sigma) \), where \( \mu \) is the mean vector and \( \Sigma \) is the covariance matrix. In what follows, \( x(t) \) is the sequence of observed states up to time \( t \), i.e. \( x(t) := (x_1, \ldots, x_t) \) and \( u(t) \) is the sequence of observed inputs up to time \( t \), i.e. \( u(t) := (u_1, \ldots, u_t) \). Also, we define the system dataset (up to time \( t \)) as \( d(t) = (x_0, u_1, x_1, \ldots, u_t, x_t) \).

Finally, we recall that, given two pdfs, say \( f_1 \) and \( f_2 \), over the same set, say \( S^* \), the Kullback-Leibler (KL) divergence \( D(f_1 || f_2) \) \cite{15} is defined as

\[
D(f_1 || f_2) := \int f_1(s^*) \ln \left( \frac{f_1(s^*)}{f_2(s^*)} \right) ds^*.
\]

We also recall that, see e.g. \cite{2} and references therein, the KL divergence measures the proximity of the pair of pdfs \( f_1 \) and \( f_2 \).

Remark 1: Applications where the evolution of a system of interest is described via a pdf naturally arise in the context of artificial intelligence and machine learning. For example, certain Bayesian reinforcement learning schemes (such as Thompson Sampling) keep a joint posterior distribution for the belief of the model. Then, the policy is derived to select optimal actions with respect to this posterior \cite{16}. Another example is provided by closed loop systems where the underlying process is classified via a neural network. See e.g. \cite{17}, \cite{18}, \cite{19} for an application in the context of intelligent transportation.

Mathematical background

Assume a system model (specified in the probabilistic sense) is given, together with the ideal probability distribution

\[
\hat{f}(d(t), x(t)|x_0) = \prod_{t \in t^*} \hat{f}(x_t|u_t, x_{t-1}) \hat{f}(u_t|d(t-1)),
\]

denoted in what follows by \( \hat{f}(\hat{\cdot}) \) for simplicity. The ideal distribution is the pdf corresponding to the desired behavior of the system. As shown in \cite{11}, the pdf can be constructed so as to embody both a given set of control goals and constraints. Indeed, in \cite{11} the pdf \( \hat{f}(x_t|u_t, x_{t-1}) \) specifies the desired state evolution and \( \hat{f}(u_t|d(t-1)) \) specifies the constraints on the evolution of the control law over time.

In \cite{11} an algorithm has been proposed with the goal of producing a control distribution, \( \hat{f}(u_t|d(t-1)), \ t \in t^* \), such that the joint probability distribution

\[
f(d(t)|x_0) = \prod_{t \in t^*} f(x_t|u_t, x_{t-1}) \hat{f}(u_t|d(t-1)),
\]

(denoted for simplicity by \( f(\tilde{\cdot}) \)) minimizes the Kullback-Leibler divergence \( D(f(\tilde{\cdot}) || \hat{f}(\cdot)) \). In what follows we simply say that \( \hat{f}(u_t|d(t-1)) \) is an admissible control strategy (or policy) for the system. The main theoretical result for the design of the control strategy, which has been formalized in \cite{11} (see Proposition 2), can be stated as follows:

**Theorem 1:** The optimal admissible control strategy \( \hat{f}(u_t|d(t-1)) \) minimizing \( D(f(\tilde{\cdot}) || \hat{f}(\cdot)) \) is the randomized control strategy:

\[
\hat{f}(u_t|d(t-1)) = \frac{\hat{f}(u_t|d(t-1))e^{-\omega(u_t,d(t-1))}}{\gamma(d(t-1))},
\]

where, starting from \( \gamma(d(t)) = 1, \gamma(\cdot) \) and \( \omega(\cdot, \cdot) \) are computed via the following system of backward recursive equations:

\[
\begin{align}
\gamma(d(t-1)) &= \int \hat{f}(u_t|d(t-1))e^{-\omega(u_t,d(t-1))} du_t, \quad (4a) \\
\omega(u_t, d(t-1)) &= \int f(x_t|x_{t-1}) \log \left( \frac{\hat{f}(x_t|x_{t-1})}{\gamma(d(t)) \hat{f}(x_t|x_{t-1})} \right) dx_t. \quad (4b)
\end{align}
\]

**Problem statement**

Theorem \cite{11} is obtained by assuming that the pdf of the model is known to the control. Unfortunately, this assumption cannot be satisfied in certain applications. For example, in the context of Deep Learning, neural networks can only provide an estimate of the underlying pdf dominating the process that is being observed. Motivated by this, in this paper we present a sensitivity analysis for the above randomized control strategy to errors in the system model, i.e. errors in \( f(x_t|x_{t-1}) \). In particular, by focusing on the case where the pdf of the model and the ideal pdf are both Gaussian, we first perform an analysis of model error propagation and then we draw some considerations on the safety regions of the closed loop system (this concept is introduced in Section III-D) and on how iterative learning techniques can be integrated in the control strategy so as to extend the safety region.

III. Results

In this section we present the main results of the paper. We also illustrate the results with a number of simulations.

A. Randomized policy: the Gaussian case

Assume that: (i) the pdf of the system is Gaussian, in particular \( f(x_t|x_{t-1}) = \mathcal{N}(A_t x_{t-1} + B_t u_t, \Sigma_t) \); (ii) the ideal distributions are also Gaussian distributions, i.e. \( \hat{f}(x_t|u_t, x_{t-1}) = \mathcal{N}(\hat{\mu}_t, \hat{\Sigma}_t) \) and \( \hat{f}(u_t|d(t-1)) = \mathcal{N}(\hat{\mu}_t, \hat{\Sigma}_t) \). With the following proposition an explicit formula is given for the control strategy (see also \cite{8}).

**Proposition 1:** The optimal admissible control strategy \( \hat{f}(u_t|d(t-1)) \) minimizing \( D(f(\tilde{\cdot}) || \hat{f}(\cdot)) \) is the randomized control strategy given by the system of recursive equations

\[
\begin{align}
\gamma(d(t-1)) &= Ce^{-\frac{1}{2}(L_{t-1} x_{t-1} + M_{t-1})^T \Sigma_{t-1}^{-1} (L_{t-1} x_{t-1} + M_{t-1})}, \quad (5a) \\
\gamma(d(t-1)) &= Ce^{-\frac{1}{2}(L_{t-1} x_{t-1} + M_{t-1})^T \Sigma_{t-1}^{-1} (L_{t-1} x_{t-1} + M_{t-1})}, \quad (5b)
\end{align}
\]

with \( \gamma(d(t)) = 1, C \) being a constant not depending on the data and where the control parameters are given by the following set of backward recursive equations:
where $\mu_{t,u}$, $\Sigma_{t,u}$ can be devised from the recursive equations (5b) and assume that (5b) is satisfied for some $t$. Note that (4b) can be rewritten as:

$$\omega(u_t, d(t-1)) = D(f(x_t|u_t, x_{t-1}))[l f(x_t|u_t, x_{t-1})] - \int f(x_t|u_t, x_{t-1}) \log(\gamma(d(t))) dx_t.$$ (7)

Since $f(x_t|u_t, x_{t-1})$ and $f(x_t|u_t, x_{t-1})$ are both multivariate normal distributions, and since $x_t$ is $n$-dimensional, it can be shown that:

$$D(f(x_t|u_t, x_{t-1}))[l f(x_t|u_t, x_{t-1})] = \frac{1}{2} \left( \log \frac{1}{|\Sigma_t|} \right) - n + tr(\Sigma_t^{-1} \Sigma_t)$$

$$+ (\mu_x - l \mu_{t,x})^T \Sigma_t^{-1} (\mu_x - l \mu_{t,x})$$ (8)

where $\mu_x = A_t x_{t-1} + B_t u_t$. Since (5b) is satisfied at time $t$, then computing the second term of the right hand side of (7) yields

$$E[\log(\gamma(d(t)))] = C - \frac{1}{2} E[(L_t X_t + M_t)^T \Sigma_t^{-1}(L_t X_t + M_t)]$$

$$= C' - \frac{1}{2} (L_t \mu_x + M_t)^T \Sigma_t^{-1}(L_t \mu_x + M_t)$$ (9)

where $C' = C - \frac{1}{2} tr(\Sigma_t^{-1} L_t \Sigma_t L_t^T)$. It can be shown that, by combining (8) and (9), by completing the squares and normalizing by $\gamma(d(t-1))$, gives (6f) and (6g)

Moreover, combining (9) with (4b) and (4a) gives us (6a), (6b), (6c), (6d), (6e). □

Remark 2: In Proposition 1 we considered a model described by $f(x_t|u_t, x_{t-1}) = N(A_t x_{t-1} + B_t u_t, \Sigma)$, it is known that a random vector $x_t \sim N(A_t x_{t-1} + B_t u_t, \Sigma)$ is given by the stochastic equation

$$x_t = A_t x_{t-1} + B_t u_t + \xi_t,$$

where $A_t$ is the time-varying $n \times n$ state matrix, $B_t$ is the time-varying $n \times m$ control matrix and $\xi_t \sim N(0, \Sigma)$ is some noise.

Remark 3: The recursive equations given in Proposition 1 imply that $x_t$ is distributed following a multivariate normal. In turn, this allows to explicitly compute the mean vector and the covariance matrix for the pdf describing $x_t|d(t-1)$.

Example: the optimal control strategy

We now illustrate the effectiveness of the control strategy of Proposition 1 via a representative example. In particular, we consider a single input-single output system. The system model is given by:

$$f(x_t|u_t, x_{t-1}) = N(ax_{t-1} + bu_t, \Sigma),$$ (10)

where $a = 1.27$, $b = 0.04$, $\Sigma = 0.6$. The ideal distributions are instead:

$$f(x_t|u_t, x_{t-1}) = N(0, l \Sigma_t),$$

$$f(u_t|d(t-1)) = N(0, l \Sigma_t)$$

where $l \Sigma_t = 0.2$ and $l \Sigma_t = 0.4$. Finally, we set the time horizon to $t = 100$. Note that the uncontrolled system is unstable (a is indeed greater than 1). However, the above result implies that the optimal admissible policy of Proposition 1 is able to stabilize the system and, moreover, ensure that $x_t$ will be distributed in accordance to the specified pdf $f(x_t|u_t, x_{t-1})$. This is confirmed in Figure 1.

The figure has been obtained by running 1e5 simulations and then by averaging, for each $t$ in the time horizon, the results across all the realizations. Finally, in Figure 2 we further characterize the statistical distribution of $x_t$ by binning the variable, in order to approximately visualize the underlying pdf.

B. Link with the Linear Quadratic Regulator

In the special case when $l \mu_{t,x} = l \mu_{t,u} = 0$, the recursive equations in Proposition 1 become the well-known update equations as those of the Linear Quadratic Regulator when the following cost function is minimized:

$$J = x_t^T l \Sigma_t^{-1} x_t + \sum_{t=0}^{t-1} x_t^T l \Sigma_t^{-1} x_t + u_t^T l \Sigma_t^{-1} u_t.$$

Therefore all the results for the LQR apply in this set-up and the infinite horizon case will involve solving the same Algebraic Riccati Equation. Interestingly, this allows to recover the Certainty Equivalence Principle from the Linear Quadratic-Gaussian control [20].

Additional considerations can be drawn on the mean and variance of $x_t$. For the sake of notational ease, assume that
The time evolution of \( x_t \) can be seen as the solution of the stochastic equation

\[
x_t = Ax_{t-1} + B\mu_{t,u} + \xi_t + \xi_{u,t},
\]

where \( \xi_{u,t} \sim \mathcal{N}(0, \Sigma_{t,u}) \). Then:

\[
\mu_{t,x} = (A - B\Sigma_{t,u}B^T\Sigma_{t,w}^{-1}A)\mu_{t-1,x} = K\mu_{t-1,x},
\]

\[
\Sigma_{t,x} = \Sigma_{\Xi} + K\Sigma_{t-1,x}K^T + B\Sigma_{t,u}B^T.
\]

In particular, since the controller is randomized, from (12b) we see that the variance of the controller (which is still the optimal controller in the KL-Divergence metric) affects the variance of the state variable. This is the major difference between the randomized control and the LQG or LQR control algorithms. In such algorithms, the control variable is chosen deterministically once \( x_{t-1} \) is known. This observation leads us to the following proposition.

**Proposition 2:** Assume that the probabilistic control given by the equations in Proposition 1 is used to control the system (both in variance and in mean) are the same conditions as in the ones of the LQR. Namely, that all the eigenvalues of the matrix \( S = (A - B\Sigma_{t,u}B^T\Sigma_{t,w}^{-1}A) \) lie inside the unit circle.

The proof of this result is omitted here for the sake of brevity.

**C. Analysis of model error propagation**

Now, we turn our attention to study how sensitive the control algorithm in Proposition 1 is with respect to errors in the models. In doing so, we consider the infinite horizon setting, i.e. when \( t \to +\infty \). Also, we assume that \( \Sigma_{t,u}, \Sigma_{t,x} \) are constant and \( \mu_{t,u} = \mu_{t,x} = 0 \). Moreover, in the context of this analysis, the real statistical model for the system is given by:

\[
f(x_t|u_t, x_{t-1}) = \mathcal{N}(Ax_{t-1} + Bu_t, \Sigma),
\]

while the control algorithm of Proposition 1 does not have access to the real model, but only to its approximation. Specifically, the control algorithm has access to the approximated system model

\[
\hat{f}(x_t|u_t, x_{t-1}) = \mathcal{N}(\hat{A}x_{t-1} + \hat{B}u_t, \hat{\Sigma}_{t,\xi}). \tag{13}
\]

**Remark 4:** Situations where the control algorithm has only access to a statistical model that can only approximate the real model of the system naturally arise in many applications. For example, it is known that, in the context of deep learning, neural networks [2] can be used to approximate the pdf of the underlying process being classified. In other applications involving iterative learning, the presence of bias in the training datasets can lead to biased models which could considerably affect the outcomes of the learning process, see e.g. [21]. With the analysis present here we seek to investigate the impact of model approximations on the control algorithm of Proposition 1.

Given the set-up of this Section, the recursive equations for the control algorithm of Proposition 1 are considerably simplified. This is formalized with the following proposition.

**Proposition 3:** Consider the control algorithm of Proposition 1 and assume that: (i) \( t \to +\infty \); (ii) \( \Sigma_{t,u}, \Sigma_{t,x} \) are constant; (iii) \( \mu_{t,u} = \mu_{t,x} = 0 \); (iv) the approximate model used by the algorithm is given in (13). Then, the randomized control strategy is given by the system of recursive equations

\[
\Sigma_{\omega}^{-1} = \frac{f \Sigma_{\omega}^{-1} + \hat{\Sigma}_{\omega}^{-1} \hat{B}}{f - \hat{B} \hat{\Sigma}_{\omega}^{-1} \hat{A}},
\]

\[
\mu_{t,u} = -\Sigma_{\omega}^{-1} \hat{B} \Sigma_{\omega}^{-1}A x_{t-1},
\]

\[
\Sigma_{\omega}^{-1} = \Sigma_{\omega}^{-1} (\Sigma_{\omega}^{-1} - \hat{\Sigma}_{\omega}^{-1} \hat{B} \hat{\Sigma}_{\omega}^{-1} \hat{A})^{-1} \hat{B} \Sigma_{\omega}^{-1} \hat{A}.
\]

where \( \Sigma_{\omega}^{-1} \) solves the algebraic Riccati equation

\[
\Sigma_{\omega}^{-1} = \frac{f \Sigma_{\omega}^{-1} + \hat{B}^T \Sigma_{\omega}^{-1} \hat{A}}{f - \hat{A} \Sigma_{\omega}^{-1} \hat{B}}.
\]
The proof of the above result can be obtained via direct inspection, by using the assumptions of Proposition 3 to simplify the equations given in Proposition 1.

From the above proposition note that the randomized policy can be computed by solving a Riccati equation. That is, in order to study stability of the system we need to study the eigenvalues of the matrix \((A - BS_uB^T \Sigma_u A)\). Now, consider the case where the system controlled by the algorithm of Proposition 3 is stable. That is, we assume that \(\rho(A - BS_uB^T \Sigma_u A) < 1\) where \(\rho(M)\) is the spectral radius of the matrix \(M\). Clearly, when \(A \neq \tilde{A}\) and \(B \neq \tilde{B}\) then such a condition can be violated. We are interested in identifying for which set of parameters, \(A\) and \(B\), stability is preserved. This leads to the following:

**Definition 1:** The region of convergence of a system controlled by the algorithm of Proposition 3 is the region of the parameters space
\[
K_{A,B} = \{(A, B) : \rho(A - BS_uB^T \Sigma_u A) < 1\}.
\]

Since \(\rho(\cdot)\) is continuous, \(K_{A,B}\) is an open set that includes at least one point in the space parameter, \((A, B)\). Therefore it must contain a neighborhood of that point. Furthermore, the region of convergence must necessarily be bounded in the \(\tilde{A}\) direction for a fixed \(\tilde{B} \neq 0\) if \(\rho(A) < 1\) then \((\tilde{A}, 0) \in K_{A,B}\) for all \(A\). The behavior with respect to \(\tilde{B}\) is more difficult to analyze in the matrix case due to the implicit definition of \(\Sigma_u^{-1}\). We can, however, find the boundary of \(K_{A,B}\) using an algorithm for continuation of an implicitly defined manifold, for example [22]. These algorithms need an initialization point on the boundary. In this case, a point where \(\rho(A - BS_uB^T \Sigma_u A) = 1\) or close enough. The point can be found using any root finding algorithm.

The above observations imply that the region of convergence can be numerically computed via continuation methods if some points of the parameter space are known to belong to the region. Suppose we have some theoretical knowledge on \(A\) and \(B\), namely, we are given a subset \(\tilde{K}\) of the parameter space where we know the system is stable. We are interested now in computing the set:
\[
K_K = \cap_{(A, B) \in K} K_{A, B}.
\]

Alternatively, one can define this set as
\[
K_K = \{(\tilde{A}, \tilde{B}) : \max_{(A, B) \in K} \rho(A - BS_uB^T \Sigma_u A) < 1\}.
\]

Again, the function \(\max_{(A, B) \in K} \rho(A - BS_uB^T \Sigma_u A)\) is continuous and therefore we can compute the boundary \(\max_{(A, B) \in K} \rho(A - BS_uB^T \Sigma_u A) = 1\) using the algorithm in [22]. Thus, we have the following proposition, stating a condition ensuring computability of the convergence region.

**Proposition 4:** Given theoretical knowledge of a compact set \(K\), we can compute the set \(K_K\), guaranteeing that any \(\tilde{A}, \tilde{B} \in K_K\) will make the system stable.

**Example (continued): effects of model errors and convergence region**

Consider, again, the system model (10). This time, we consider the infinite horizon control problem and assume that the control algorithm does not have a complete knowledge of the model but rather it only has access to an approximate version, namely:
\[
\tilde{f}(x_t|u_t, x_{t-1}) = N(ax_{t-1} + \tilde{b}u_t, \Sigma_x).
\]

where \(\tilde{b} = 0.02\). That is, the only difference of the approximate model with respect to the real system model lies in the parameter \(b\). As shown in Figure 3 and Figure 4, this mismatch leads to significant changes in the behavior of the closed loop system, when the control algorithm of Proposition 3 is used.

Since the system is scalar, we were able to solve explicitly the Riccati equation in order to estimate the region of convergence of the control algorithm. In particular, once we found one point in the parameter space for which the closed loop system converged (left panel of Figure 5) we were able to estimate the region \(K_K\), in accordance to Proposition 4.

**D. Safety of the closed loop system**

Given the set-up of Section III-C we now consider safety of the closed loop system, when the control algorithm of Proposition 3 is used to control a system of interest, for which only the approximate model (13) is known. Intuitively, inspired by e.g. [23], we define the safety constraints of the closed-loop system in terms of a safety region, i.e. a region of the state space that is, with high probability, forward invariant. Essentially, if the initial conditions of the system start in the safety region, then this region, with high probability, is never left by the system.
The case where the initial conditions, \( x_0 \), the safety condition as a function of trajectories. In this paper we are interested in characterizing region of convergence estimated via continuation methods \([22]\). point of convergence, obtained by solving the Riccati equation. Right panel: which the system converges (in yellow, in both panels). Left panel: initial Fig. 5. For a given region in which the parameters lie, we find a region in which the system converges (in yellow, in both panels). Left panel: initial point of convergence, obtained by solving the Riccati equation. Right panel: region of convergence estimated via continuation methods \([22]\).

trjectories. In this paper we are interested in characterizing the safety condition as a function of \( \hat{A} \) and \( \hat{B} \). We consider the case where the initial conditions, \( x_0 \), of the system are safe and specify the safety region in terms of \( x_0 \). This is formalized with the next definition, where \( \| \cdot \| \) denotes Euclidean norm:

**Definition 2:** Let \( M > 0 \), \( \delta > 0 \) and let \( x_0 \) be the initial system conditions. The \((M, \delta)\)-safety region for the closed loop system is the region of the parameter space

\[
K_{A,B}(M, \delta) = \{ \hat{A}, \hat{B} : \mathbb{P}(\|x_t\| > M\|x_0\|) < \delta \ \forall t \}.
\]

**Remark 5:** A major difference of Definition 2 with respect to the one given in [4], [23] is that the \((M, \delta)\)-safety region is a region of the system parameter space, rather than the system state space.

The following proposition is an analogous of Proposition 4 and ensures the computability of the safety region.

**Proposition 5:** Assume that the initial conditions, \( x_0 \), of the system are safe. Then, the safety region given in Definition 2 can be computed for any \( \delta > 0 \) and \( M > 0 \).

Essentially, since the distribution of \( x_1 \) is controlled, the set \( K_{A,B}(M, \delta) \) is, again, computable via the algorithmic continuation approach of e.g. \([22]\).

**Remark 6:** In principle, computing the sets \( K_K \) and \( K_{K}(M, \delta) \) allows to select appropriate \( \hat{B}, \hat{A} \) under uncertainty in the model \( A, B \). This might be of use in a system identification scenario, when the parameters of the identified system are uncertain but bounds on uncertainty are known.

E. Integrating iterative learning of the model

Suppose now that we are in a scenario in which the system model is learned via an iterative learning process (see e.g. \([1]\) for an application in the context of model predictive control). From the viewpoint of the probabilistic control analyzed in this paper, we are not interested in designing the learning strategy. Rather, we model the learning process by having \( A_t, B_t \) such that:

\[
\| A - \hat{A}_t \| \leq \varepsilon_1 + \varepsilon_2 t,
\]

and

\[
\| B - \hat{B}_t \| \leq \varepsilon_3 + \varepsilon_4 t,
\]

where \( \varepsilon_2 \) and \( \varepsilon_4 \) are smaller than one. Intuitively, with the above equations we model the case where the learning process allows to update the matrices \( A_t \) and \( B_t \) so that such matrices get closer to the real ones. Also, note that we consider the non-ideal case, where we allow for the learning process to convergence to matrices that are close to the real ones. This is modeled by the fact that, when \( t \to +\infty \), we have \( \| A - \hat{A}_t \| \leq \varepsilon_1 \) and \( \| B - \hat{B}_t \| \leq \varepsilon_3 \).

Given the scenario illustrated above, it can be shown (the proof will be presented elsewhere) that the (stable) control policy can be computed as follows:

1) assume that, at each time step, \( t \), the matrices \( \hat{A}_t \) and \( \hat{B}_t \) are equal to the real matrices;
2) assume that such matrices will not change over time;
3) apply (13a) and (13b) on such matrices.

Finally, the introduction of a learning mechanism modeled as described above, also improves the characterization of the safety region. In particular, it can be shown that the region can be computed and, moreover, it is possible to compute the set of initial conditions for which the system is safe.

The numerical methods, which are still based on the use of continuation algorithms, are omitted here and will be presented elsewhere.
Example (continued): safety region

Finally, we considered again the approximate model in (16). This time, we assume that the parameter \( \hat{b} \) changes over time and it indeed converges exactly to \( b \). In this case, we were able to compute the \((3, 0.1)\)-safety region of the system, as shown in Figure 6.

IV. CONCLUSIONS

We presented a sensitivity and a safety analysis for the so-called fully probabilistic control scheme. In particular, after presenting the sensitivity analysis, we focused on characterizing the convergence region of the closed loop system and introduced a safety analysis. Finally, we discussed how the introduction of learning mechanisms can be beneficial for both convergence and safety.

ACKNOWLEDGMENTS

BGP was partially supported by the NSF grant number 1514606. GR was supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number 16/RC/3872.

REFERENCES

[1] U. Rosolia and F. Borrelli, “Learning model predictive control for iterative tasks: a data-driven control framework,” IEEE Transactions on Automatic Control, vol. 63, no. 7, pp. 1883–1896, July 2018.
[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
[3] R. S. Sutton and A. G. Barto, Introduction to Reinforcement Learning, 1st ed. Cambridge, MA, USA: MIT Press, 1998.
[4] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, “Safe model-based reinforcement learning with stability guarantees,” in Advances in Neural Information Processing Systems 30, 2017, pp. 908–918.
[5] B. Recht, “A tour of reinforcement learning: The view from continuous control,” CoRR, vol. abs/1806.09460, 2018.
[6] K. J. Astrom, Introduction to stochastic control theory. Academic Press New York, 1970.
[7] J. Berger, Statistical Decision Theory and Bayesian Analysis. Springer, 1984.
[8] M. Krn, “Towards fully probabilistic control design,” Automatica, vol. 32, no. 12, pp. 1719–1722, 1996.
[9] M. Karny, Optimized Bayesian Dynamic Advising: Theory and Algorithms (Advanced Information and Knowledge Processing). Berlin, Heidelberg: Springer-Verlag, 2005.
[10] M. Krn, J. Bhm, T. V. Guy, and P. Nedoma, “Mixture-based adaptive probabilistic control,” International Journal of Adaptive Control and Signal Processing, vol. 17, no. 2, pp. 119–132.
[11] M. Krn and T. V. Guy, “Fully probabilistic control design,” Systems & Control Letters, vol. 55, no. 4, pp. 259–265, Apr. 2006.
[12] R. Herzallah, “Fully probabilistic control for stochastic nonlinear systems with input dependent noise,” Neural Networks, vol. 65, pp. 199 – 207, 2015.
[13] ——, “Generalised probabilistic control design for uncertain stochastic control systems,” Asian Journal of Control, vol. 0, no. 0.
[14] A. G. Barto, R. S. Sutton, and C. W. Anderson, “Neuronlike adaptive elements that can solve difficult learning control problems,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, no. 5, pp. 834–846, Sept 1983.
[15] S. Kullback and R. Leibler, “On information and sufficiency,” Annals of Mathematical Statistics, vol. 22, pp. 79–87, 1951.
[16] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, “Bayesian reinforcement learning: A survey,” Found. Trends Mach. Learn., vol. 8, no. 5–6, pp. 359–483, Nov. 2015.
[17] S. Sweeney, R. Ordez-Hurtado, F. Pilla, G. Russo, D. Timoney, and R. Shorten, “A context-aware e-bike system to reduce pollution inhalation while cycling,” IEEE Transactions on Intelligent Transportation Systems, pp. 1–12, 2018.
[18] Y. Lassoued, J. Monteil, Y. Gu, G. Russo, R. Shorten, and M. Mevisen, “A hidden markov model for route and destination prediction,” in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Oct 2017, pp. 1–6.
[19] S. Sinott, R. Ordez-Hurtado, G. Russo, and R. Shorten, “On the design of a route parsing engine for connected vehicles with applications to congestion management systems,” in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Nov 2016, pp. 1586–1591.
[20] R. Stengel, Optimal control and estimation. New York: Dover, 1984.
[21] H. v. Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, ser. AAAI’16. AAAI Press, 2016, pp. 2094–2100.
[22] M. E. Henderson, “Multiple parameter continuation: computing implicitly defined k-manifolds,” International Journal of Bifurcation and Chaos, vol. 12, no. 03, pp. 451–476, Mar. 2002. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218127402004498
[23] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, “Learning-based model predictive control for safe exploration,” in Proceedings of the 2018 IEEE Conference on Decision and Control, 2018 (to appear).