MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Kernel methods and Gaussian processes for system identification and control

Care, Algo; Carli, Ruggero; Dalla Libera, Alberto; Romeres, Diego; Pillonetto, Gianluigi

TR2023-112 September 02, 2023

Abstract

This article reviews some kernel-based approaches for system identification and learning-based control. In the first part, the presentation moves from classic linear system identification, to nonlinear system identification in Reproducing Kernel Hilbert Spaces. The kernel-based regularization methods are illustrated in a tutorial manner. Moreover, the probabilistic (Bayesian) interpretation of kernels is also introduced, with focus on the Gaussian Processes (GPs) framework, a special case of great practical interest. The second part touches upon the problem of quantifying the uncertainty of the estimated dynamic systems from different points of views (deterministic, probabilistic, probabilistic and robust). The final part of the article surveys the applications of GPs in robust control, adaptive control, model predictive control, feedback linearization and reinforcement learning.

IEEE Control Systems Magazine 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Kernel methods and Gaussian processes for system identification and control

A ROAD MAP ON REGULARIZED KERNEL-BASED LEARNING FOR CONTROL

{Algo Carè, Ruggero Carli, Alberto Dalla Libera, Diego Romeres, and Gianluigi Pillonetto}
POC: G. Pillonetto (giapi@dei.unipd.it)
June 7, 2023

Summary

This article reviews some kernel-based approaches for system identification and learning-based control. In the first part, the presentation moves from classic linear system identification, to nonlinear system identification in Reproducing Kernel Hilbert Spaces. The kernel-based regularization methods are illustrated in a tutorial manner. Moreover, the probabilistic (Bayesian) interpretation of kernels is also introduced, with focus on the Gaussian Processes (GPs) framework, a special case of great practical interest. The second part touches upon the problem of quantifying the uncertainty of the estimated dynamic systems from different points of views (deterministic, probabilistic, probabilistic and robust). The final part of the article surveys the applications of GPs in robust control, adaptive control, model predictive control, feedback linearization and reinforcement learning. Three GP-based algorithms (PILCO, Black-Drops, and MC-PILCO) are tested on the swing-up task for a simulated cart-pole and experiments on a real Furuta pendulum conclude the article.

Digital Object Identifier 10.1109/MCS.2020.000000
Date of current version: XXXXXX

The commonly adopted route to control a dynamic system, and make it follow the desired behaviour, consists of two steps. First, a model of the system is learnt from input-output data, a task known as system identification in the engineering literature. Here, an important point is not only to derive a nominal model of the plant but also confidence bounds around it. The information coming from the first step is then exploited to design a controller that should guarantee a certain performance also under the uncertainty affecting the model. This classical way to control dynamic systems has been recently subject of new intense research thanks to an interesting cross-fertilization with the field of machine learning. New system identification and control techniques have been developed with links to function estimation and mathematical foundations in Reproducing Kernel Hilbert Spaces and Gaussian processes. This has become known as the *Gaussian regression (kernel-based) approach to system identification and control*. It is the purpose of this article to give an overview of this development.

The goal of System Identification is to build mathematical models of dynamic systems from input-output measurements. This term was introduced in 1953 by Lotfi Zadeh and, starting from the seminal paper [1], such discipline has become a fundamental subfield of Automatic Control with many years of theoretical achievements and a remarkable impact on engineering applications. There are many textbooks available on the subject like [2], [3]. They describe identification techniques based on paradigms from mathematical statistics. In particular, the classical approach relies on prediction error methods (PEM) [4] and the concept of discrete model order. Different architectures are postulated, each of them parametrized by an unknown finite-dimensional parameter vector θ. If the true system has the postulated model structure, and the noise is Gaussian, this procedure is asymptotically optimal: it cannot be outperformed by any other unbiased estimator as the number of measurements grows to infinity [5]. However, a crucial point here is the selection of the most adequate model structure. In this classical framework, complexity of the different structures is typically connected with the number of their unknown parameters. Determining the dimension of θ then involves a trade-off between bias and variance: the model should be flexible enough to describe the experimental data but not too complex to be fooled by noise. This can be carried out by using complexity measures, such as Akaike's information criterion (AIC) [6], the Bayesian formulation (BIC) [7], minimum description lenght (MDL) [8], [9] or cross validation (CV) [10], [11]. A graphical illustration is in the left part of Fig. 1. In the linear and time-invariant setting, the model structures $\mathcal{M}$ there depicted could e.g. represent Finite-dimensional Impulse Response (FIR) models. Each FIR of length d is associated to a d-dimensional parameter vector θ whose components are the impulse response coefficients. Other fundamental structures are the rational transfer functions where the Laplace transform of the impulse response is modeled as the ratio of two polynomials with unknown coefficients contained in θ. In both of these examples, as the dimension of θ goes to infinity, the model becomes so complex that it can approximate any kind of impulse response [12].

The alternative route to system identification overviewed in this paper is illustrated in the right part of Fig. 1. Instead of postulating finite-dimensional models of increasing complexity, the system is directly searched for in a high-dimensional space. In the linear setting, the space $\mathcal{H}$ can e.g. contain all the FIR models of fixed and large dimension d. One could also set $d = \infty$ to obtain the space with all the possible impulse responses. In this way, system identification becomes an ill-posed inverse problem in the sense of Hadamard [13]: an infinite-dimensional object has to be inferred from a finite set of input-output measurements. The challenge is now to control model complexity without necessarily reducing the model dimension. A powerful way to restore well-posedness is to regularize the problem by introducing a suitable ranking of possible solutions over $\mathcal{H}$. Among many different systems able to describe the experimental data in a similar way, the one that most reflects our expectations is selected. For instance, in the estimation of linear and Bounded Input Bounded Output (BIBO) systems, impulse responses that smoothly decay to zero should be privileged. In the nonlinear setting, where stability is a more delicate concept with several facets [14], [15], [16], one could promote input-output relationships that are smooth (similar inputs provide similar outputs) and embed fading memory concepts (as the lag increases, past inputs are expected to be less influent on the output). Regularization techniques can be used to introduce the desired ordering of solutions for inverse problems. In particular, the scope of the regularizers is to include in the estimation process useful information on the function/dynamic system to be reconstructed. In this survey, we focus on Kernel-based methods, and their Bayesian interpretation leading to Gaussian processes [17]. A fundamental feature of this approach is that the space $\mathcal{H}$ and the ranking over it (assigned by means of function norms), can be defined just specifying a positive-semidefinite kernel. This is a map that enjoys the same properties of the covariance function in probability theory [18]. It induces a particular space, called reproducing kernel Hilbert space (RKHS) [19], containing functions whose properties are strictly related to those of the kernel. For instance, absolutely integrable kernels are especially useful for linear and BIBO dynamic systems since they induce RKHSs that are stable, i.e. that contain only absolutely integrable impulse responses [20].

Once the kernel is assigned, the system estimate can be obtained as the solution of an optimization problem containing two competing terms: adherence to experimental data and a penalty term accounting for the ranking induced by the kernel. These two components have to be balanced by the so called hyperparameters which need to be tuned using data. An important example is the regularization parameter (a positive scalar denoted by γ in the optimization problem reported in the right part of Fig. 1) which can be tuned in a continuous manner. Hence, it defines (in some sense) a continuous-model order, enriching the system identification problem with a whole new dimension. The automatic selection of such hyperparameters has proved to be a powerful and versatile approach compared to the classical rules of choosing model discrete orders [21], [20]. As already mentioned, kernel-based approaches also enjoy an important stochastic interpretation where the ranking is seen as the manifestation of a Bayesian prior placed over a space of systems [22], [23], [24]. If the system is modeled a priori as a (often zero-mean) Gaussian random field with covariance equal to the kernel, when data become available the kernel-based estimator provides its minimum variance estimate. We will see that this has important consequences for kernel selection, hyperparameters estimation and computation of the uncertainty affecting the system estimate.

The key reason of the renewed interest of control community towards regularization has been the introduction of new kernels/covariances that account for dynamic systems features. A large variety is now available in the linear setting [25], [26], [27], [28], [29]. They are derived in deterministic settings, working also over the frequency domain [30], [31], or in a stochastic framework, e.g. exploiting maximum entropy concepts [32], [33], [34]. Furthermore, the theory of RKHSs of stable impulse responses has been recently developed [35], [36], [37]. Many open issues still remain in the nonlinear case but powerful kernels are available even in this context, e.g. the Gaussian and Matern kernel, common in machine learning literature [22], and the polynomial one, related to classical (truncated) Volterra models [38], [39] which describe systems through a large set of monomials [40], [41], [42], [43], [44]. Their applications regarding e.g. mechanical systems can be found in [45], [46], [47], [48], [49], [50]. Kernels allow here to encode many basis functions in an implicit way, maintaining the problem computationally tractable.

Like all the estimators, kernel-based estimators must be accompanied by a rigorous evaluation of their error. In general, error bounds can be obtained in deterministic, [51], or probabilistic set-ups, [2], [3]. Probabilistic set-ups are particularly suitable to strike a satisfactory balance between the conservatism of a bound and the risk of it being wrong. Moreover, thanks to the already mentioned link between kernels and Bayesian priors, it is possible to study the error bounds in a fully Bayesian framework, where the uncertain model is itself treated as a stochastic quantity, [52]. For Gaussian probabilities, such a Bayesian approach is not only theoretically sound but also computationally tractable, with a large impact on applications, [22]. It should however be noticed that the ensuing error bounds are sensitive to the choice of the probabilistic description of the uncertainty, a choice that is largely left to the user's discretion (not to say whim). Hence, a present challenge is developing probabilistic approaches that are robust, in the sense that they remain valid for large classes of probability distributions (e.g., all the distribution indexed by some hyperparameters, [53], [54]), and yet are informative in spite of the large dimension of $\mathcal{H}$. As we shall see, both classic and recent system identification literature can be a source of inspiration in the pursuit of this goal.

The building of mathematical models and uncertainty bounds around them described above is an important step to control dynamic systems. Overall, this leads to the so called model-based control methods (they are opposed to direct data driven techniques that allow to tune a controller, belonging to a given class, without the need of an identified model of the system). In particular, in the last decade a great effort has been devoted to the design of learning-based control combining kernel-based methods/Gaussian processes (GPs) with robust control for linear systems [55], [56], [57], adaptive control [58], [59], feedback linearization [60], [61], and Model Predictive Control (MPC) [62], [63], [64]. The prediction models that are inferred using machine learning techniques include also other approaches different from GPs, like deep neural networks (DNNs). The advantage of using deep learning such as feedforward neural networks, convolutional neural networks, and long short-term memory networks lies on the ability of abstracting large volumes of data and enabling real-time execution in a control loop. On the other hand, Gaussian regression requires higher computational burden for larger sets of data but provide uncertainty bounds around the model that can be naturally incorporated into traditional control frameworks. GPs are in fact used either to identify the overall dynamics, or to assess a residual model uncertainty to be added to a known nominal model. More recently, GPs framework has been adopted also in model-based reinforcement learning (RL) algorithms for control purposes. It is known that RL is often not so data efficient, i.e., it requires many trials to learn a particular task. This makes RL methods often largely inapplicable to mechanical systems that quickly wear out. Typically, model-based methods, i.e., methods that learn a dynamics model of the environment, are more apt to extract valuable information from experimental data than model-free methods. Hence, they are more promising to increase data efficiency. In [65], the authors have introduced a model-based policy search method, called PILCO (probabilistic inference for learning control) where a GP framework has been employed to learn a probabilistic dynamical model and to explicitly incorporate the model uncertainty into the long-term planning. The predicted distributions are approximated as Gaussians by using exact moment matching, thus allowing policy evaluation in closed form and analytic calculations of gradient for policy improvements. PILCO has been shown to be able to cope efficiently with little data and to improve learning from scratch in only a few trials. Extensions of PILCO along different directions have been provided in [66], [67], [68], [69], [70].

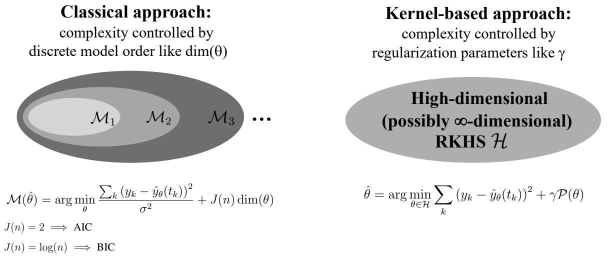
Classical approach:
complexity controlled by
discrete model order like dim(θ)
$\mathcal{M}_1$
$\mathcal{M}_2$
$\mathcal{M}_3$
...
$\mathcal{M}(\hat{\theta}) = \arg\min_{\theta} \frac{\sum_k (y_k - \hat{y}_\theta(t_k))^2}{\sigma^2} + J(n)\dim(\theta)$
$J(n) = 2 \implies$ AIC
$J(n) = \log(n) \implies$ BIC
Kernel-based approach:
complexity controlled by
regularization parameters like γ
High-dimensional
(possibly ∞-dimensional)
RKHS $\mathcal{H}$
$\hat{\theta} = \arg\min_{\theta\in\mathcal{H}} \sum_k (y_k - \hat{y}_\theta(t_k))^2 + \gamma\mathcal{P}(\theta)$

Figure 1 *Classical approach to system identification (left)* Different finite-dimensional model structures $\mathcal{M}_i$ are postulated, each of them inducing a different output predictor $\hat{y}$ parameterized by θ. Complexity is regulated by a term J that may depend on the data set size n and by the discrete model-order, typically established by the dimension of the vector θ. Commonly used criteria are AIC or BIC. *Kernel-based approach to system identification (right)* The unknown system is searched for in a high-dimensional space space, e.g. a reproducing kernel Hilbert space, whose dimension can be also infinite. Complexity is tuned in a continuous way by means of hyperparameters like the regularization parameter γ which has to balance adherence to experimental data and a penalty $\mathcal{P}$ assigned to θ. The penalty can e.g. include information on smoothness of the input-output map and on system stability.

In light of this Introduction, this Survey will be naturally divided in three main parts. First, we will describe how modern regularization theory may return accurate models of linear and nonlinear dynamic systems. Then, we will discuss how to complement them with informative (non-asymptotic) uncertainty bounds. Then, we will see that GPs (kernel technology) can be fruitfully exploited for robust control purposes. Examples involving simulated and real data will be included to describe the practical implications of the methodology here described.

KERNELS AND GAUSSIAN REGRESSION: AN INTRODUCTION TO SOME KEY CONCEPTS

Consider the problem of estimating an unknown function f from direct and noisy samples. The measurements model is

$$y_i = f(x_i) + e_i, \quad i = 1, \dots, n. \tag{1}$$

where the x_i are often called input locations while e_i are stochastic additive noises. One classical approach to estimate the function is to assume that f is the linear combination of known basis functions ϕ_i through unknown coefficients α_i. The model of dimension d is so given by

$$f(x) = \sum_{i=1}^{d} \alpha_i \phi_i(x),$$

where also d has typically to be learnt from data. This is a crucial aspect of the regression problem: good estimates of f require a trade-off between adherence to experimental data and model complexity. Note that any subspace spanned by $\{\phi_i\}_{i=1}^d$ may represent the model structure $\mathcal{M}_d$ depicted in Fig. 1 (left panel).

For illustrative purposes a numerical experiment is now introduced. The unknown function is defined over the unit interval $[0,1]$ and reported in Fig. 2 (top left) together with 100 measurements corrupted by white Gaussian noise. We adopt the following sinusoidal basis functions

$$\phi_i(x) = \sqrt{2}\sin(x(i\pi - \pi/2)). \tag{2}$$

For any d, the archetypical approach to determine $\{\alpha_i\}_{i=1}^d$ is *least squares*, i.e. the one that minimizes the squared error between the observed outputs and those predicted by the model:

$$\arg\min_{\{\alpha_i\}} \sum_{j=1}^{n} \left(y_j - \sum_{i=1}^{d} \alpha_i \phi_i(x_j)\right)^2.$$

To determine d, in this experiment we will use an *oracle* which knows the true function. Among the different least squares estimates obtained (one for any different choice of d), it selects that estimate which maximizes the fit

$$100\left(1 - \frac{\|\hat{f} - f\|}{\|f\|}\right), \quad \|\cdot\| = \text{Euclidean norm}. \tag{3}$$

For this data set, the oracle selects $d = 9$ basis functions leading to a fit around 80%. The estimate is reported in Fig. 2 (top right) and appears quite close to truth. However, we will try to improve such result through the different approach illustrated in Fig. 1 (right panel) which suggests to directly start from a high-dimensional space. For this purpose, we fix $d = 100$, a dimension which equals the number of available measurements. The challenge now is to control complexity without using d and making use of regularization. This means that, among solutions that describe the data in a similar way, the regularizer should favour those that mostly agree with our expectations on f. One of the first regularized approaches proposed in literature is ridge regression [71], [72]. In our context, it determines the unknown function obtaining the expansions coefficients as

$$\arg\min_{\{\alpha_i\}} \sum_{j=1}^{n} \left(y_j - \sum_{i=1}^{d} \alpha_i \phi_i(x_j)\right)^2 + \gamma \sum_{i=1}^{d} \alpha_i^2 \tag{4}$$

with $n = d = 100$. In (4), the additional term $\sum_i \alpha_i^2$ which complements least squares is the regularizer while the positive scalar γ is the so called regularization parameter. It has to trade-off the data fit and the penalty term and can be seen as the (continuous) counterpart of (the discrete order) d in the regularized setting. In our experiment, we tune γ still using the oracle: the fit (3) is now function of γ and the oracle maximizes it. The oracle-based estimate is reported in Fig. 2 (bottom left) and appears worse than that returned by the classical approach. Indeed, the fit is only 57%. It would seem that the idea to start from a high-dimensional space has no advantages. But the crucial point is that we need to improve the regularizer. To this regard note from (2) that, as i increases, the power of the basis functions ϕ_i is concentrated at higher frequencies. Many functions (systems) encountered in nature have however some regularity properties: one expects that their energy decays at higher frequencies. This information can be encoded by introducing in the regularizer some weights ζ_i that decay to zero. Specifically, *generalized ridge regression* determines the expansion coefficients as

$$\hat{\alpha} = \arg\min_{\{\alpha_i\}} \sum_{j=1}^{n} \left(y_j - \sum_{i=1}^{d} \alpha_i \phi_i(x_j)\right)^2 + \gamma \sum_{i=1}^{d} \frac{\alpha_i^2}{\zeta_i}. \qquad (5)$$

The meaning of the new regularizer $\sum_i \alpha_i^2/\zeta_i$ is that not all the basis functions should be treated in the same way: as i increases more penalty has to be assigned to α_i and, hence, to ϕ_i. For our specific example, we still set $n = d = 100$ and use

$$\zeta_i = (i\pi - \pi/2)^{-2} \qquad (6)$$

(the rationale underlying this choice will be clear a few lines below). In addition, we now estimate γ using only the data y_i (no oracle is used) with an approach which will be revealed at the end of this section. The estimate is reported in Fig. 2 (bottom right) and appears close to truth, leading to a fit around 94%: the new estimator (without the use of the oracle) outperforms the (oracle-based) classical approach.

Generalized ridge regression as kernel-based regularization

The key to interpret (5) as a regularized kernel-based estimator is to build a (positive definite) *kernel*: it embeds basis functions ϕ_i and weights ζ_i as follows

$$\mathcal{K}(x, x') = \sum_{i=1}^{d} \zeta_i \phi_i(x) \phi_i(x'). \qquad (7)$$

Fixing x and seeing the kernel as function of x', the kernel section centred on x is obtained. As explained in this survey, the theory based on reproducing kernel Hilbert spaces [19], coupled with the famous representer theorem [73], then ensures that the estimate (5) can be equivalently written in terms of n kernel sections, where n is the data set size. In other words, if $\hat{\alpha}_i$ come from (5), one has

$$\begin{aligned} \hat{f}(x) &= \sum_{i=1}^{d} \hat{\alpha}_i \phi_i(x) \\ &= \sum_{i=1}^{n} \hat{c}_i \mathcal{K}(x, x_i) \end{aligned}$$

where the coefficients $\hat{c}_i$ solve a linear system. This leads to some fundamental facts:

- » instead of formulating basis functions and weights, it can be convenient to formulate directly a kernel which implicitly encodes them;
- » the modeling process thus finds a new dimension since kernel properties encode our expectations on f. For instance, smooth kernels promote smooth function estimates, integrable kernels ensure integrable function estimates.

To further appreciate these points, it is now useful to reveal the rationale underlying the choice of the basis function and weights adopted to solve our regression problem. Using (2) and (6), then letting the dimension d go to infinity, the associated kernel becomes [74]

$$\begin{aligned} \mathcal{K}(x, y) &= \sum_{i=1}^{+\infty} \frac{2\sin(x(i\pi - \pi/2))\sin(y(i\pi - \pi/2))}{(i\pi - \pi/2)^2} \\ &= \min(x, y). \end{aligned} \qquad (8)$$

This is the famous spline kernel [73] and permits to reformulate the estimator (5) as optimization over a function (Sobolev) space, obtaining the smoothing spline estimator:

$$\hat{f}(x) = \arg\min_{f} \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2 + \gamma \int_0^1 \dot{f}^2(x)dx. \qquad (9)$$

We can now give the estimate reported in Fig. 2 (bottom right) a different and important interpretation: it describes the experimental evidence trying also to minimize the energy of the first-order derivative of f. One can now wonder which model could be used if the unknown function is expected to be more regular. Instead of introducing new complicated/mysterious basis functions and weights, one can just increase kernel regularity. For example, the second-order spline kernel is smoother than (8) and penalizes the energy of the second-order derivative [73]. The Gaussian kernel (introduced later on) is the most used in machine learning: it is very smooth and returns estimates differentiable for all degrees [75].

Kernel-based regularization as Gaussian regression

The estimate obtained by ridge regression and reported in Fig. 2 (bottom left) appears unsatisfactory since it contains oscillations perceived as unrealistic. Before seeing the data, more regularity is expected. Under the deterministic setting so far adopted, where f is an unknown *deterministic* function, the spline kernel $\mathcal{K}(x, y) = \min(x, y)$ improves the result since it induces the penalty term $\int \dot{f}^2(x)dx$, hence favouring smoother profiles. This same penalty can be also given a stochastic interpretation. In fact, it is interesting now to note that the spline kernel (8) corresponds exactly to the covariance of an important *stochastic process* known as Brownian motion (integrated white Gaussian noise) [18]. To establish a connection with the kernel-based estimator (9), it is needed now to think of f as a zero-mean Gaussian process over the unit interval of covariance $\min(x, y)$. This means that, for any integer m, the function f evaluated over any set $[x_1\ x_2\ \ldots\ x_m]$ is a zero-mean Gaussian vector with $\mathcal{E}(f(x_i)f(x_j)) = \min(x_i, x_j)$ ($\mathcal{E}$ indicates mathematical expectation). This stochastic description of the unknown f includes smoothness information: the Brownian motion is continuous with probability one. Such information is encoded in the probability density function p of any random variable $f(x)$. It is called prior since it describes the uncertainty affecting the function before seeing any measurement. After seeing the output data contained in the vector Y, such probability density can be updated according to the Bayes' rule, becoming the posterior density

$$\mathrm{p}(f(x)|Y) \propto \mathrm{p}(Y|f(x))\mathrm{p}(f(x))$$

where $\mathrm{p}(Y|f(x))$ is the likelihood function. This permits also to define the posterior mean $\mathcal{E}(f(x)|Y)$ which corresponds to the minimum variance estimate of $f(x)$ given Y [76]. The link with the kernel-based estimator $\hat{f}$ in (9) now arises. If the data Y are corrupted by white Gaussian noise, for a suitable choice of γ, one has $\mathcal{E}\left[f(x)|Y\right] = \hat{f}(x)\ \forall x$. As we will see, this fact is not related to the particular (spline) kernel illustrated in this introductory example. It holds for any kernel, over any possible domain, once $\mathcal{K}$ is seen as the covariance of a zero-mean *Gaussian random field f*. Two important advantages arise:

- the regularization parameter, as well as variables like noise variances and kernel parameters, are often unknown. The stochastic interpretation of kernel-based regularization permits to use statistical criteria to tune them. We can now reveal that the estimate in Fig. 2 (bottom right) was obtained by estimating γ via maximization of the so called *marginal likelihood* given by $\mathrm{p}(Y|\gamma)$ and widely described later on;
- after setting the regularization parameters to their estimates, Gaussian uncertainty bounds around the estimates can be easily computed being available in closed-form. This point will be especially important for *robust control* purposes.

The unknown functions encountered in this survey

Unknown functions appear in several problems related to identification of dynamical systems. In particular, in *nonlinear system identification f* can represent the unknown input-output map with the input locations x_i that contain past input (and possibly also output) data. The dimension of x_i is related to the system memory. A special case arises in the linear setting where each f is linear in x, so that we can write $f(x) = \theta^T x$. Here, θ is a vector which contains unknown impulse response coefficients. Regularizers introduced in this setting rely on linear kernels $\mathcal{K}(x,y) = x^T P y$ where P is a symmetric semidefinite positive matrix. Assuming P invertible, such kernels lead to penalty terms of the form $\theta^T P^{-1}\theta$.

Other unknown functions encountered in the survey arise in *state-space models* and are related to state transitions and output measurements equations. Just focusing e.g. on state transitions, in discrete-time one has

$$x_{t+1} = \mathbf{f}(x_t) + \text{noise} \tag{10}$$

where t now denotes time. So, each component of $\mathbf{f}$ is an unknown scalar function f evaluated at input locations x_t defined sequentially by the evolution of the states.

We start describing the linear and time-invariant case in the following section.

FROM CLASSICAL TO KERNEL-BASED LINEAR SYSTEM IDENTIFICATION

We consider a single-input single-output (SISO) linear and time-invariant discrete-time dynamic system. Its unknown impulse response is denoted by g with components $\{g_k\}_{k=1}^{+\infty}$. The noisy outputs are

$$y_i = \sum_{k=1}^{+\infty} u_{i-k} g_k + e_i, \quad i = 1, \dots, n \tag{11}$$

where u is the known input and e_i are the measurement noises. The latter are assumed independent, zero-mean with variance σ^2. Our goal is to estimate g from knowledge of u and the n measurements y_i.

It is apparent that linear system identification corresponds to inverting a convolution operator. This problem is also known as deconvolution and is ubiquitous in biology, physics and engineering [77], [78]. It is difficult since convolution in discrete- and also continuous-time is a well-behaved operator but its inverse may not exist or may be unbounded [79]. Indeed, impulse response estimation is an intrinsically ill-posed problem because (11) requires to reconstruct an infinite number of coefficients g_k from a finite number of observations.

In this section, we will briefly overview some impulse response estimators that restore well-posedness within the setting of classical and kernel-based system identification. In general, it will be useful to measure the estimation performance in terms of mean squared error (MSE) and impulse response fit. In this regard, let $\|\cdot\|$ be the Euclidean or ℓ_2 norm, e.g.

$$\|g\|^2 = \sum_{k=1}^{+\infty} g_k^2.$$

An estimator $\hat{g}$ of g is a random object since it depends on the input-output measurements (the outputs y_i are random variables since they are affected by stochastic noise). One has

$$\begin{aligned} MSE_{\hat{g}} &= \mathcal{E}\|\hat{g} - g\|^2 \\ &= \underbrace{\sum_{k=1}^{+\infty} \mathcal{E}(\hat{g}_k - \mathcal{E}\hat{g}_k)^2}_{Variance} + \underbrace{\sum_{k=1}^{+\infty} (g_k - \mathcal{E}\hat{g}_k)^2}_{Bias^2}, \end{aligned} \tag{12}$$

where the error has been decomposed in the last passage into two components. The first one is the *variance* while the difference between the mean and the true impulse response defines the *bias*. Often, complex models of dynamic systems lead to estimators with low bias but large variance. If the mean coincides with g, the estimator is said to be *unbiased*.

When data become available, the realization of $\hat{g}$ becomes our impulse response estimate. We will then define the fit as

$$FIT_{\hat{g}} = 100\Big(1 - \frac{\|\hat{g} - g\|}{\|g\|}\Big). \tag{13}$$

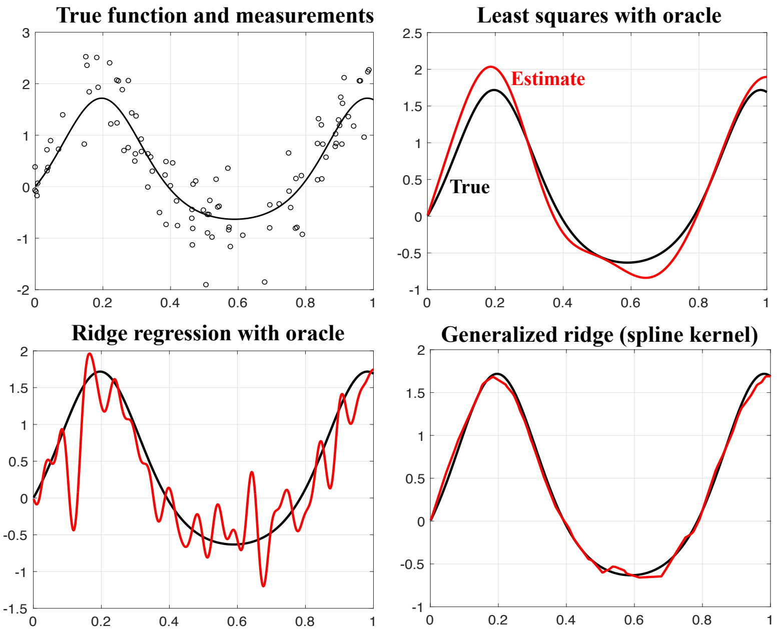

Figure 2 *Function regression problem* True function and noisy samples (top left), estimate from least squares with sinuosidal basis functions and an oracle to select their number (top right), from ridge regression with oracle to select the regularization parameter (bottom left) and from kernel-based spline regression with regularization parameter estimated via marginal likelihood (bottom right)

Hence, values close to 100 indicate that the $\hat{g}$ is a very accurate reconstruction of the linear system.

Classical approach

As already anticipated, the classical approach relies on the introduction of a family of model structures of different complexity, see the left part of Fig. 1. In this linear setting, each structure is a collection of impulse responses that we indicate with g_θ. They are parametrized by a deterministic vector θ and may contain a different number $\dim(\theta)$ of parameters. The simplest example is given by the FIR models where well-posedness is restored by assuming that the impulse response contains only a finite number of nonzero coefficients. For different dimensions d, each FIR is characterized by the d-dimensional vector θ containing the unknown impulse response coefficients, i.e.

$$g_\theta = \theta_1, \qquad \dim(\theta) = 1 \qquad (19a)$$

$$g_\theta = \begin{pmatrix} \theta_1 & \theta_2 \end{pmatrix}^T, \qquad \dim(\theta) = 2 \qquad (19b)$$

$$\vdots \qquad (19c)$$

Hence, (11) can be rewritten in matrix form in terms of the following linear regression problem

$$Y = \Phi\theta + E \qquad (20)$$

where Y and E are n-dimensional (column) vectors whose i-th components are, respectively, y_i and e_i. Furthermore, Φ is the $n \times d$ regression matrix whose i-th row is

$$\begin{pmatrix} u_{i-1} & u_{i-2} & \dots & u_{i-d} \end{pmatrix}.$$

The least squares estimate of θ is

$$\theta^* = \arg\min_\theta \|Y - \Phi\theta\|^2 = (\Phi^T\Phi)^{-1}\Phi^T Y \qquad (21)$$

where, for simplicity, Φ is assumed of full column rank. However, this usually leads to an ill-conditioned problem: even small errors in the measurements can lead to a large estimation error. Ill-conditioning may be severe also when d is just set to that value able to well capture system dynamics. As an example, if the system is stable and d is sufficiently large, in practice (20) holds exactly if θ contains the first d components of the true f. Hence, the least squares estimator is virtually unbiased and the MSE in (12) reduces to the trace of the matrix $\sigma^2(\Phi^T\Phi)^{-1}$. In presence of ill-conditioning, the matrix $\Phi^T\Phi$ is close to singularity so that the trace of the inverse can be very large. This shows that FIR models are easy to fit to data but they can suffer of large variance. In addition, the variance worsens when the system input is a low-pass signal. This is a situation often encountered in real applications that, when data are realizations from stationary stochastic processes, admits a spectral characterization via the Szegő theorem (which studies the asymptotic behaviour of large Toeplitz matrices) [80], [81].

Structures g_θ can be defined by using other building blocks, e.g. the Laguerre basis functions [82] defined in the z-transfer domain by

$$H_i(z) = \frac{(1-\alpha z)^{i-1}}{(z-\alpha)^i}, \quad -1 < \alpha < 1, \quad i = 1, 2, \dots \qquad (22)$$

Linear and time-invariant dynamic systems

Any discrete-time dynamic system can be seen as a mapping from an input sequence to an output sequence. The system is Linear and Time-Invariant if the such mapping is linear and does not depend on actual time. A sequence of zeros with only one element different from zero is an Impulse and defines an output called the Impulse Response (IR). IR determines the output for any input since any sequence decomposes as a linear combination of impulses happening at different time instants. A linear and time-invariant dynamic system is Bounded Input Bounded Output (BIBO) stable if any bounded input produces a bounded output. This property is equivalent to requiring that the IR be absolutely summable. In discrete-time this means that the IR should decay to zero sufficiently fast.

For a Single-Input Single-Output (SISO) time-invariant linear systems, a general model structure depending on an unknown parameter vector θ is defined by the transfer functions F mapping inputs contained in u to outputs and the transfer function G mapping a white noise e to an output additive disturbance. Let us consider one time unit as sampling interval and use q to indicate the shift operator $qu(t) = u(t+1)$ (one could also use the complex variable z in place of q to formulate the next equations using the z-transform). Then, it holds that

$$y(t) = F(q,\theta)u(t) + G(q,\theta)e(t) \qquad \text{(S14a)}$$

$$\mathcal{E}e^2(t) = \sigma^2; \quad \mathcal{E}e(t)e(k) = 0 \text{ if } k \neq t \qquad \text{(S14b)}$$

where $\mathcal{E}$ indicates mathematical expectation. The IRs of the system are then given by the expansion of $F(q,\theta)$ and $G(q,\theta)$ in the inverse (backwards) shift operator:

$$F(q,\theta) = \sum_{j=1}^{\infty} f(j,\theta)q^{-j} \qquad \text{(S15)}$$

$$G(q,\theta) = 1 + \sum_{j=1}^{\infty} g(j,\theta)q^{-j}. \qquad \text{(S16)}$$

If $G = 1$, an output error (OE) model is obtained as in (11).

Popular black-box linear models (no physical insight) use parametrizations with F and G rational in the shift operator:

$$F(q,\theta) = \frac{B(q,\theta)}{H(q,\theta)}; \quad G(q,\theta) = \frac{C(q,\theta)}{D(q,\theta)} \qquad \text{(S17)}$$

where B, H, C, D are all polynomials of q^{-1} with the (unknown) polynomial coefficients contained in the parameter vector θ. Typically, the dimension of θ, i.e. the polynomials order, need to be estimated from data.

Letting $H = D$ the important *ARMAX models* are obtained [2]. Another fundamental case is $H = D$ and $C = 1$ which gives the *ARX model*:

$$y(t) = B(q)u_k + (1 - D(q))y + e(t) \qquad \text{(S18)}$$

Finally, $C = D$ leads to $G = 1$, i.e. the (already mentioned) OE model, now with rational deterministic transfer function F (as in (23) in terms of z-transform). Furthermore, if $H = 1$, F reduces to a single polynomial in q^{-1}: the impulse response has a finite number of nonzero coefficients and one obtains the FIR models (19).

where α regulates the decay rate of the impulse response. Note that the case $\alpha = 0$ makes us come back to FIR models. Other important descriptions used to better balance the variance and bias components illustrated in (12) are the rational transfer functions where the z-transform of the impulse response is the ratio of two polynomials:

$$\frac{a_{d_1}z^{d_1} + a_{d_1-1}z^{d_1-1} + \ldots + a_0}{z^{d_2} + b_{d_2}z^{d_2-1} + \ldots + b_0}, \quad d_1 \leq d_2. \qquad (23)$$

In any case, assigned the dimension d, we can first introduce the loss function

$$V(\theta) = \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{+\infty} u_{i-k} g_{\theta,k} \right)^2 \qquad (24)$$

and then we can obtain θ through PEM, i.e. solving the following nonlinear least squares problem

$$\theta^* = \arg\min_{\theta} V(\theta) \qquad (25)$$

that generalizes (21). Problem (25) coincides with the maximum likelihood procedure if the noises are Gaussian, independent and with the same variance. In many real world problems, the dimension d of θ is however unknown and must be determined from data. This problem is key since the choice of model complexity will have a major effect on the quality of the final model.

Cross validation (CV) is widely used for model order selection [10]. Once impulse responses estimates of different dimension are obtained by (25), CV tries to select the one with the largest prediction capability on future data. Holdout validation is the simplest version of CV: the available measurements are split into two sets. The first one is the *training set* and is used to train the model. The other one is the *validation set* and is exploited to evaluate the prediction capability. Thanks to its nature, CV may be applied to the most varied situations.

The so called Akaike-like criteria are also popular to determine model complexity and do not require to divide the data in different partitions. To illustrate them, for the sake of simplicity, let the measurement noise be white and Gaussian of variance σ^2. Then, for known σ^2, the "optimal" model minimizes

$$\left[\frac{V(\theta^*)}{\sigma^2} + J(\dim(\theta), n) \right] \quad \text{known } \sigma^2 \qquad (26)$$

while, if σ^2 is unknown and included in θ, the objective becomes

$$[n \log(V(\theta^*)) + J(\dim(\theta), n)] \quad \text{unknown } \sigma^2 \qquad (27)$$

The penalty

$$J(\dim(\theta), n) = 2\dim(\theta) \quad \text{AIC} \tag{28}$$

leads to the well known Akaike's criterion (AIC) [6] which, for large samples, gives an approximately unbiased estimator of the Kullback-Leibler divergence (the distance of a model from the true data generator). A larger penalty on model flexibility, derived following Bayesian arguments, is instead defined by

$$J(\dim(\theta), n) = \log(n)\dim(\theta) \quad \text{BIC} \tag{29}$$

and is called Akaike's criterion-type B, BIC, or Rissanen's Minimum Description Length (MDL) criterion [8], [7], [2]. One limitation of AIC and BIC is that all of these criteria are based on an approximation of the likelihood that is only asymptotically exact. This undermines the applicability of the theory when the ratio $n/\dim(\theta)$ is not large enough, see [20] for illustrations of these phenomena in linear system identification.

Numerical experiment using the classical approach with an oracle

Let us consider the following system identification problem. The unknown transfer function is

$$\frac{z^2+2z+1}{(z-0.8)(z-0.6)} + \frac{z^2+2z+1}{z^2-0.7z+0.7}. \tag{30}$$

The system, initially at rest, is fed with a low-pass input u given by the realization of white Gaussian noise of unit variance filtered by $1/(z-0.9)$. Note that the pole 0.9 is quite close to the unit circle, hence decreasing the power of the signal at high-frequencies and increasing the ill-conditioning affecting the identification problem. The impulse response has to be estimated from 1000 output measurements corrupted by white Gaussian noise. The signal-to-noise ratio (SNR), i.e., the ratio between the variances of the noiseless output and the noise, is 20 and the input-output data are plotted in Fig. 3. We assume that the system identification procedure is equipped with an oracle which is an estimator with access to the test data. This means that structures of different dimensions are fitted to test data using e.g. PEM and then the oracle selects the one maximizing the fit (13) which is computed using the first 50 impulse response coefficients. This procedure is ideal, not implementable in practice, but it is useful since it provides an upper bound on the performance.

First, FIR models are used. The choice of the FIR length is a trade-off between bias (a large d can be needed to represent slowly decaying impulse responses without large error) and variance (large d leads to estimation of many parameters, hence increasing the variance). To balance these two components, (21) is computed for different dimensions of θ and then the oracle selects $d = 17$ to optimize the fit which turns out 81.2%. The true impulse response and the FIR estimate are visible in Fig. 4 (left panel). The size of the training data is quite large and the SNR is not small but the FIR estimate is not satisfactory. This is due to the low-pass input that gives poor excitation and makes the problem ill-conditioned.

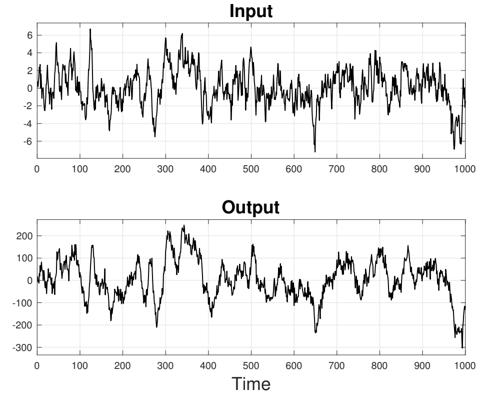

Figure 3 Input-output data generated using system (30).

To improve the results we can now resort to rational transfer functions (23). Eq. (25) is now computed for different orders and the oracle determines 4 as the optimal order of the rational transfer function. The fit increases to 87.3% and the impulse response estimate is displayed in Fig. 4 (right panel).

Regularized least squares

The result reported in Fig. 4 (left panel) would seem to suggest that, at least in presence of ill-conditioning, FIR models are not useful even when an oracle is used to select their dimension. But now let us consider a different approach where, inspired by the right part of Fig. 1, the estimate is directly searched for in a high-dimensional space, e.g. given by high-order FIR models in the linear setting. For large d, we have seen that system identification turns often out an ill-conditioned (and possibly ill-posed) problem. So, how can we control the variance without discrete tuning of d? For this purpose, one important approach is to add a regularization term to the least squares criterion. As already recalled in the previous introductory section on kernel methods to illustrate (4), the first method proposed in the literature to deal with numerical stability problems in the inversion of some operators is ridge regression [71], [72]. The estimate is given by

$$\hat{\theta} = \arg\min_{\theta} \|Y - \Phi\theta\|^2 + \gamma\|\theta\|^2 \tag{31}$$

and thus optimizes an objective which is sum of two terms. The first one is a quadratic loss function that measures adherence to experimental data. Without any other term the objective would correspond to (21). The second term is a penalty given by the squared Euclidean norm whose aim is to reduce the oscillations that can affect the least squares estimate. There is also a third very important ingredient which is a positive scalar γ, the so called regularization

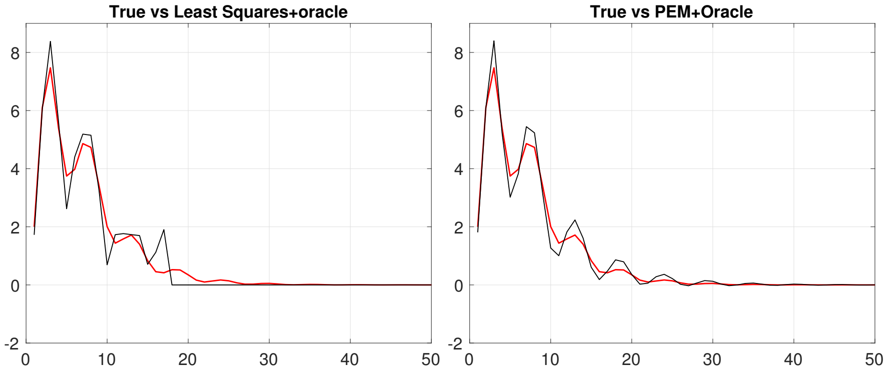

Figure 4 *Classical approach to linear system identification* True impulse response (thick red line) and estimates using FIR models (left) and rational transfer function (right) with model order selected by an oracle

parameter already encountered in (5). It has to trade off experimental evidence and the regularizer, hence balancing bias and variance. It can be seen as the counterpart of the dimension d of theta. We can now reconsider the previous numerical experiment using $d = 50$ and adopting (31) with an oracle to optimize γ. The best possible ridge estimate is reported in Fig. 5 (top left panel) and the fit is 65.5%. The reconstructed impulse response is not so satisfactory due to the presence of some unrealistic oscillations which suggest data overfitting. This drives us to generalize (31) by introducing the more sophisticated penalty $\theta^T P^{-1}\theta$ that depends on a design regularization (symmetric and positive definitive) matrix P. The following *regularized least squares* (ReLS) problem is obtained:

$$\hat{\theta} = \arg\min_{\theta} \|Y - \Phi\theta\|^2 + \gamma\theta^T P^{-1}\theta \tag{32a}$$

$$= P\Phi^T(\Phi P \Phi^T + \gamma I_n)^{-1}Y; \text{or} \tag{32b}$$

$$= (P\Phi^T\Phi + \gamma I_d)^{-1}P\Phi^T Y. \tag{32c}$$

ReLS can be also implemented using non invertible P. In (32a), one has to replace P^{-1} with the pseudo-inverse and add the constraint that the solution be orthogonal to the null space of P. In any case, the solution coincides with that reported in (32b) or (32c).

Beyond γ, the performance of ReLS crucially depends on the choice of the regularizer induced by P as already seen in the introductory example of Fig. 2. When a signal is known just to be smooth, beyond the spline kernels, one of the most used regularizers P used in the machine learning is related to the so-called Gaussian kernel. The (ij)-entry of P becomes

$$P_{ij} = \exp\left(\frac{-(i-j)^2}{\omega}\right) \tag{33}$$

where ω is the kernel width. One can think of γ and ω as knobs that may control the regularity of the impulse response. We can ask the oracle to tune them and the best possible estimate based on the Gaussian kernel is in Fig. 5 (second panel) with the fit being 83.8%. The profile is now smoother and we have improved over ridge regression. However, the peak of our impulse response is underestimated and some oscillations still affect the reconstructed profile. The Gaussian kernel does not seem the breakthrough we were hoping for. To understand the reasons, it is now useful to reconsider the MSE introduced in (12).

Assume that data are generated according to the linear regression (20) for a certain dimension d, with the true value of θ denoted by θ_0. Then, after some calculations, one obtains the following expression for the MSE of ReLS

$$\mathcal{E}[(\hat{\theta} - \theta_0)(\hat{\theta} - \theta_0)^T] =$$

$$\sigma^2\left(\frac{P\Phi^T\Phi}{\gamma} + I_m\right)^{-1}\left(\frac{P\Phi^T\Phi P}{\gamma^2} + \frac{\theta_0\theta_0^T}{\sigma^2}\right)\left(\frac{\Phi^T\Phi P}{\gamma} + I_m\right)^{-1}. \tag{34}$$

Now, we can find values of P and γ that minimize (34) in matrix sense. One obtains $\gamma = \sigma^2$ (the noise variance) with the optimal regularization matrix being [83]

$$P = \theta_0\theta_0^T. \tag{35}$$

As expected, the answer depends on the unknown θ_0. Hence, (35) cannot be used in practice but can however give some important insights on the problem. In fact, it shows that the regularization matrix P should synthesize our expectations on the impulse response. When the system is exponentially stable, the components of θ_0 will exponentially decay to zero so that also the components of P (both along and outside the diagonal) should mimic such behaviour. The first regularization matrix satisfying such requirements derives from the so called stable spline kernel [84], [21], also called TC kernel in [83]. It is defined by

$$P_{ij} = \alpha^{\max(i,j)}, \quad 0 \le \alpha < 1, \tag{36}$$

where α is a stability parameter that regulates how fast the impulse response is expected to decay to zero. Generalizations are also given by the second-order stable spline kernel [21], which increases the level of expected smoothness, and the DC kernel [83], where an additional hyperparameter is introduced to regulate the level of correlation among the samples. Many other kernels then appeared in the literature to describe linear systems, and have been mentioned in Introduction. An in-depth comparison between the classical and the kernel-based approach was proposed in [85]. The authors compared the two approaches both in terms of point estimators and of confidence intervals also obtaining that the kernel-based approach may outperform the classical approach.

We now come back to our illustrative example and ask the oracle to tune γ and α to solve our system identification problem. The resulting stable spline estimate is reported in Fig. 5 (third panel) with the fit being 93.4%. The result is now really satisfactory and outperforms also the oracle-based classical approach exploiting rational transfer functions as structures.

Linear state-space models

Several control strategies are based on state-space representations of the system evolution. In control applications, the case of linear and time invariant systems with discrete-time evolution is particularly common. In this setup, the system output at time t, hereafter denoted $y_t \in \mathbb{R}^p$, is a linear combination of the input $u_t \in \mathbb{R}^m$ and the system state $x_t \in \mathbb{R}^n$. In turn, a system of n first-order linear difference equations describes the evolution of x_t as follows:

$$\begin{aligned} x_{t+1} &= Fx_t + Gu_t + w_t \\ y_t &= Hx_t + Du_t + v_t, \end{aligned} \tag{37}$$

where $F \in \mathbb{R}^{n\times n}$, $G \in \mathbb{R}^{n\times m}$, $H \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$ are constant matrices; $w_t \in \mathbb{R}^n$ and $v_t \in \mathbb{R}^p$ are zero-mean Gaussian variables with covariance matrices R_w and R_v, which account for the process and measurement noise, respectively. Model (37) proved particularly useful for advanced control applications. For instance, remarkable results have been achieved in the case that the control objective consists in minimizing a cost function $c_t(x_t, u_t)$ quadratic w.r.t. x_t and u_t, the so called Linear Quadratic Regulator (LQR), [86], [87], also known as Linear Quadratic Gaussian (LQG) control problem in the stochastic setup [88], [89].

As regards the identification of state-space models of the kind in (37), it is appropriate to distinguish between the case that the state x_t is measurable or not. If the system state is directly observed, F and G can be estimated by solving a linear LS problem of the kind in (31) starting from $N+1$ system observations collected at time $t = 0\dots N$. Specifically, with reference to (31), $\theta \in \mathbb{R}^{n(n+m)}$ collects the F and G elements, $Y \in \mathbb{R}^{nN}$ concatenates the observed states at time $t = 1\dots N$, while the entries of the regression matrix Φ are states and inputs at time $t = 0\dots N-1$, disposed in accordance with θ and Y. The covariance of the process and measurement noise, i.e., R_w and R_v, can be estimated from the LS residuals. If some insights on F and G are available, they can be included in the estimation process through proper parametrization or regularizers. For instance, by using the the $L1$ norm as regularization in (31) instead of the $L2$ norm, sparsity of F and G is promoted.

LS identification of the state-space model cannot be performed if the states are not observable and only input-output data are available. In this case, an alternative route consists of estimating an input-output transfer function, then obtaining a state-space realization. ARX models are particularly suitable for this task. Compared to the FIR models introduced before, they include also an autoregressive part with past outputs seen as additional inputs. For the sake of simplicity, we consider the SISO case, described by the following equation:

$$\begin{aligned} y_i =& -a_1 y_{i-1} - \dots - a_{n_a} y_{i-n_a} + b_1 u_{i-1} + \dots \\ &+ b_{n_b} u_{i-n_b} + e_i = \varphi_y^T(i)\theta_a + \varphi_u^T(i)\theta_b + e_i, \end{aligned} \tag{38}$$

with $\theta_a = \begin{bmatrix} a_1 & \cdots & a_{n_a} \end{bmatrix}^T$, $\theta_b = \begin{bmatrix} b_1 & \cdots & b_{n_b} \end{bmatrix}^T$ while the column vectors $\varphi_y(i)$ and $\varphi_u(i)$ are built using y and u in an obvious way. As the orders n_a and n_b grow to infinity, ARX models can approximate any linear system [2].

Also (38) is a linear regression model, involving two regression matrices Φ_a, Φ_b whose i-th row is given by $\varphi_y^T(i)$ and $\varphi_u^T(i)$, respectively. In matrix form, we can thus write

$$Y = \Phi_a\theta_a + \Phi_b\theta_b + E := \Phi\theta + E \tag{39}$$

where $\theta = [\theta_a^T \ \theta_b^T]^T$. The same regularization ideas illustrated above can be now applied just partitioning the regularization matrix P as follows

$$P(\eta) = \begin{bmatrix} P^a(\eta_1) & 0 \\ 0 & P^b(\eta_2). \end{bmatrix} \tag{40}$$

where P^a and P^b are e.g. the stable spline/TC kernels (36).

After estimating the ARX model in (39), one can obtain a state-space realization. For instance, considering the controllable canonical form and $n_a = n_b = n$ in (39), the deterministic part of the system is described by

$$\begin{aligned} F &= \begin{bmatrix} 0 & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \\ G &= \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}^T \\ H &= \begin{bmatrix} b_n & \dots & b_2 & b_1 \end{bmatrix}^T \\ D &= 0. \end{aligned}$$

If the model dimension is too large, hence complicating the control design, n can be reduced using standard reduction techniques, e.g. the algorithms in [90], [91] implemented by the MATLAB function *balred*.

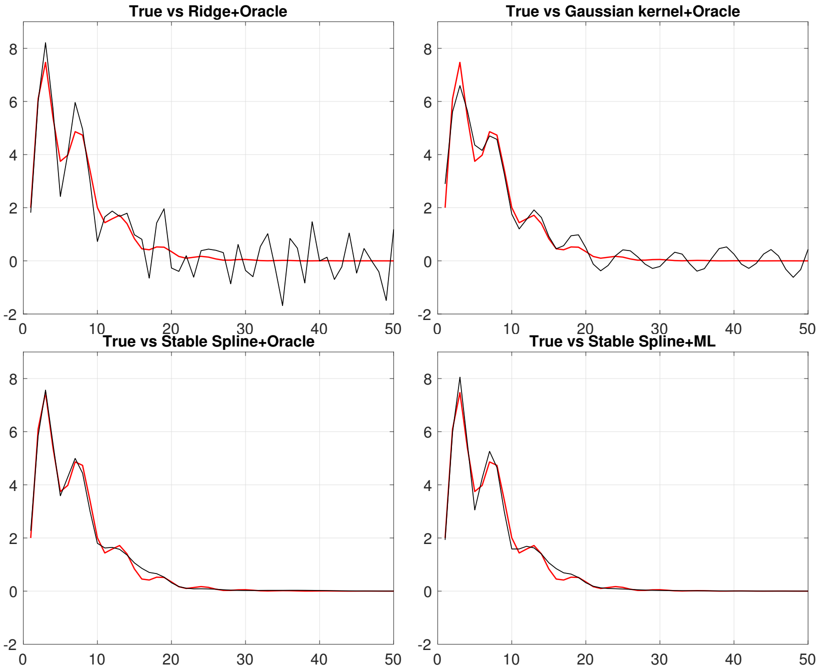

Figure 5 *Regularized linear system identification* True impulse response (thick red line) and estimates using Ridge regression with oracle (top left), Gaussian kernel with oracle (top right), stable spline with oracle (bottom left) and with hyperparameters estimated via marginal likelihood (bottom right).

Bayesian interpretation: Gaussian regression

The Bayesian interpretation of ReLS is now introduced. To simplify exposition, the FIR case is treated but the same results here exposed hold also for the ARX models introduced in the previous section e.g. see [92]. The fact that an estimate like the one reported in the top left panel of Fig. 5 is perceived as unsatisfactory suggests that there is some form of prior knowledge on the level of acceptability of candidate solutions. We have seen that this knowledge, e.g. given by system stability, guides the choice of the regularizer added to the usual sum of squared residuals. Such a design process has been described by assuming that the unknown impulse response is a deterministic vector. All the randomness of our estimators then come from the random nature of the noise. We will now see that an alternative formalization of prior information can be given by adopting a subjective/Bayesian estimation paradigm. In particular, kernel-based regularization enjoys also a stochastic interpretation where a Gaussian distribution is assigned to the impulse response. The regularization quadratic term then becomes a consequence of this prior. Let us assume that data come from the linear regression model (20) for a certain dimension d but with θ now given by a zero-mean Gaussian random vector. Its definite positive covariance is proportional to the matrix P, i.e.

$$\theta \sim \mathcal{N}(0, \lambda P) \tag{41}$$

where λ is a positive scale factor. Let θ be independent of the measurement noise which is white and Gaussian of variance σ^2. Now, we can compute the minimum variance estimate of the impulse response, i.e. the mean of the posterior of θ conditional of Y. Recall that, in view of (20), Y and θ are jointly Gaussian variables and θ conditional of Y is Gaussian too, e.g. see [76]. Hence, the mean of the a posteriori density function coincides with the maximum a posterior estimate (the maximizer of the posterior), and a simple application of Bayes' rule allows us to obtain

$$\mathcal{E}(\theta|Y) = \arg\min_{\theta} \|Y - \Phi\theta\|^2 + \frac{\sigma^2}{\lambda}\theta^T P^{-1}\theta. \tag{42}$$

This is exactly the kernel-based estimate $\hat{\theta}$ in (32a) once the regularization parameter γ is set to σ^2/λ. This also indicates that only the ratio between the scaling factors is relevant to the computation of a point estimate. Such a Bayesian view is important under several aspects. First, many times the stochastic interpretation may provide useful insights on merit and weakness of a certain model. For instance, just plotting some realizations from the prior provides an idea on the expected features incorporated in our system model. E.g., Fig. 6 plots realizations from zero-mean Gaussian vectors using covariances associated to ridge regression (white noise assumptions on the impulse response coefficients), Gaussian and stable spline kernel. Only the stable spline candidates include smooth exponential decay information. The other realizations hardly represent impulse responses of stable dynamic systems.

At higher level, the Bayesian view may inspire the construction of new priors e.g. by means of Maximum

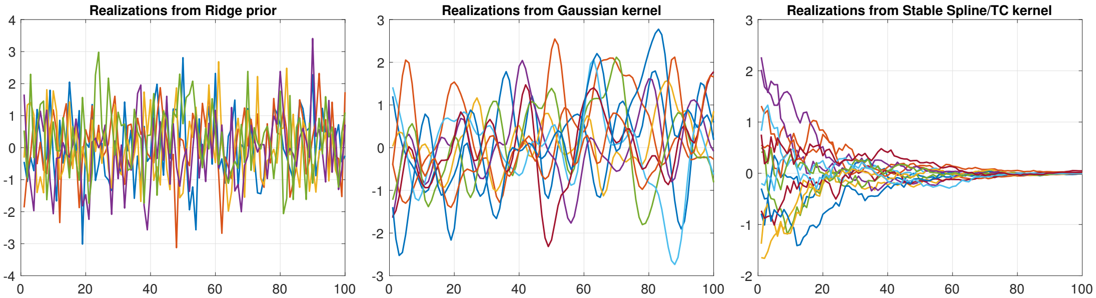

Figure 6 *Bayesian interpretation of regularization in the linear setting* Impulse responses realizations from zero-mean Gaussian vectors using covariances associated to ridge regression (left), corresponding to white noise assumptions on the coefficients, Gaussian (middle) and stable spline (right) kernel.

Entropy concepts [32]. Within this paradigm, one can derive a complete a priori density function from incomplete information, e.g. some values regarding expectations and variances. The distribution has to satisfy some constraints and maximize the entropy, hence returning, in some sense, the simplest (least committing) prior compatible with the available information. Interestingly, when this latter is just smooth exponential decay, the maximum entropy prior for θ is a zero-mean Gaussian distribution with covariance proportional to the stable spline matrix (36), see [33] for details. Other advantages of the stochastic framework are the possibility of complementing the estimates with Bayes regions (as described later on) and the derivation of statistical guidelines for hyperparameters tuning (as discussed in the next section).

Hyperparameters tuning

In real applications, the hyperparameters entering the ReLS estimator reported in (32) cannot be tuned by the oracle but have instead to be learnt from data. They include e.g. the regularization parameter γ and also some variables that define the structure of P, e.g. the kernel width ω present in (33) or the parameter α of the stable spline kernel in (36). In the following, the vector containing all the unknown hyperparameters will be denoted by η.

Many options are available to tune η. For instance, the same cross validation strategies described in the classical framework can be adopted, with the CV score now optimized w.r.t. the continuous vector η. Other important criteria do not require to split the data into a training and a validation set. A first well known class derived in a deterministic setting includes generalized cross validation and Stein's unbiased risk estimation, see [93], [11]. Another class uses the Bayesian interpretation of regularization: since the parameter γ can be seen as a noise-to-signal ratio, its estimation can be reformulated as a statistical estimation problem as described below.

In the stochastic setting our tuning problem is due to the fact that the prior on θ (and possibly on the measurement noise) is known only if we condition on η. In a fully Bayesian setting, we could also think of η as random and assign to it a prior $\mathrm{p}(\eta)$. In such a case, the prior $\mathrm{p}(\theta)$ can be computed by marginalization as $\int \mathrm{p}(\theta, \eta) d\eta$. However, in general this computation is analytically intractable. One solution is to resort to stochastic simulation to solve numerically the integral, e.g. by Markov chain Monte Carlo techniques [94]. This leads to *full Bayesian* methods. A simpler computational scheme exploits the so called *marginal likelihood* $\mathrm{p}(Y|\eta)$ which derives from the marginalization of the joint density $\mathrm{p}(Y, \theta|\eta)$ with respect to θ, i.e. $\mathrm{p}(Y|\eta) = \int \mathrm{p}(Y, \theta|\eta) d\theta$. The marginal likelihood estimate of the hyperparameters is

$$\eta^{\mathrm{ML}} = \arg\max_{\eta} \mathrm{p}(Y|\eta). \tag{43}$$

When data are sufficiently informative, one can expect $\mathrm{p}(Y|\eta)$ to be quite concentrated around η^{ML}. So, assuming the prior on η rather uninformative, the posterior can be approximated using the prior $\mathrm{p}^*(\theta) = \mathrm{p}(\theta|\eta^{\mathrm{ML}})$. In this way, the full Bayes approach is replaced by the so called *Empirical Bayes* (EB) method [95], [96], [97].

We can now specialize the EB method (43) to our high-order FIR (20). The key point is that the marginal likelihood $\mathrm{p}(Y|\eta)$ is available in closed form for any η. In fact, in view of the Gaussianity and independence of θ and E, the vector Y is zero-mean Gaussian too. One easily obtains $Y \sim \mathcal{N}(0, Z(\eta))$ with $Z(\eta) = \lambda \Phi P \Phi^T + \sigma^2 I_n$. Using the minus-log of $\mathrm{p}(Y|\eta)$, problem (43) in the context of regularized FIR becomes

$$\hat{\eta} = \arg\min_{\eta} \; Y^T (Z(\eta))^{-1} Y + \log\det(Z(\eta)). \tag{44}$$

Interestingly, the marginal likelihood may prevent overfitting. In fact, the likelihood $\mathrm{p}(Y|\eta)$ can be approximated as the product of the full likelihood and an Occam factor that penalizes unnecessarily complex systems [54], [98]. In (44) the Occam factor is represented by $\log\det(Z(\eta))$. Hyperparameters on-line tuning techniques, with new data arriving in real-time, are described in [99], [100].

To test EB we reconsider for the last time our illustrative example. Fig. 5 (bottom right panel) reports the impulse response estimate with complexity now tuned by (44). The fit turns out 91.2%, close to that of the oracle. EB, implementable in practice, outperforms the classical approach equipped with rational transfer functions and the oracle, further outlining the potentiality of regularization. Other experiments along this line can be found e.g. in [21], [83], [20].

RKHSS FOR SYSTEM IDENTIFICATION AND FUNCTION ESTIMATION

It can be tempting to set the FIR dimension to $d = \infty$, shifting the task to estimate an IIR model. But the matrix P becomes infinite so that its inverse is undefined. This makes obscure both the meaning of the regularizer $\theta^T P^{-1}\theta$ and the nature of the space where the optimizer searches for the unknown impulse response. It is then needed to generalize problem (32) by considering θ no more as a vector but as a real-valued function f defined over a generic domain $\mathcal{X}$. This operation is important since it permits to solve (in an unified framework) also other relevant problems including continuous-time linear system identification and nonlinear system identification.

To extend (32), it is key to move from positive definite matrices P to positive definite kernels $\mathcal{K}$, just called kernels in what follows, which were first encountered in the introductory section on Kernels and Gaussian regression. Given any non-empty set $\mathcal{X}$, kernels are symmetric functions over $\mathcal{X} \times \mathcal{X}$ such that, for any finite natural number p, one has

$$\sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j \mathcal{K}(x_i, x_j) \geq 0$$

for any choice of real numbers a_k and $x_k \in \mathcal{X}$. E.g., in the d-dimensional FIR case, $\mathcal{X} = \{1, \dots, d\}$ and the kernel associated with (32) is $\mathcal{K}(i,j) = P_{ij}$ for $i,j = 1, \dots, d$.

The Moore-Aronszajn theorem provides a one-to-one correspondence between $\mathcal{K}$ and particular Hilbert spaces of functions $\mathcal{H}$ known as RKHSs [19]. It contains all the finite linear combination of kernel sections, i.e. $f(x) = \sum_{i=1}^{p} a_i \mathcal{K}_{x_i}(x)$ with $\mathcal{K}_{x_i}(x) := \mathcal{K}(x_i, x)$, and some infinite combinations, i.e. the limits of Cauchy sequences w.r.t. the norm $\|f\|_{\mathcal{H}}^2 = \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j K(x_i, x_j)$. A consequence of this construction is that functions in $\mathcal{H}$ inherit properties of the kernel. Continuous kernels (also called Mercer kernels) induce spaces containing only continuous functions. Kernels that are absolutely integrable belong to the class of *stable kernels* which induce the so called *stable RKHSs* that contain only absolutely summable impulse responses. Their complete characterization can be found in [36], [37]. For instance, the stable spline kernel (36), extended to the entire set of natural numbers, is

$$\mathcal{K}(i,j) = \alpha^{\max(i,j)}, \quad i,j = 1, \dots, \infty. \tag{45}$$

This kernel can be proved to be positive definite and one also has the property $\sum_{ij} |\mathcal{K}(i,j)| = \sum_{ij} \alpha^{\max(i,j)} < \infty$. The associated RKHS thus contains (possibly infinite) combinations of exponentially decaying functions and all of them are impulse responses of BIBO stable systems. These examples, like the estimation results reported in Fig. 2, convey an important message for modeling. In place of introducing a set of basis functions to describe f, like those of Laguerre (22), in the RKHS setting one has just to choose a kernel that encodes the desired properties of the function to be estimated. Another key RKHS feature that was already described in (7) is that a continuous kernel over $\mathcal{X} \times \mathcal{X}$ admits over any compact domain the following Mercer expansion

$$\mathcal{K}(x,z) = \sum_{i=1}^{d} \zeta_i \rho_i(x) \rho_j(z), \quad x, z \in \mathcal{X} \tag{46}$$

(if $d = \infty$, and the ρ_i are mutually independent, the space is infinite-dimensional). Then, it can be proved that the basis functions ρ_i span all the RKHS and that

$$f(x) = \sum_{i=1}^{d} a_i \rho_i(x) \implies \|f\|_{\mathcal{H}}^2 = \sum_{i=1}^{d} \frac{a_i^2}{\zeta_i}.$$

Many times, the expansion (46) is not even available in closed form: kernels thus allow to use in an implicit way a possibly infinite number of basis functions.

This simple introduction to RKHSs already allows us to generalize (32). Any measurement y_i entering (32) is a noisy version of the linear transformation of the vector θ given by $\Phi(i,:)\theta$ where $\Phi(i,:)$ is the i-th row of the regression matrix. In the RKHS setting, θ is replaced by the function f over $\mathcal{X}$ and its transformation is denoted by $L_i[f]$ with L_i representing a linear and continuous functional. For instance, if f denotes a continuous-time impulse response, $L_i[f]$ can now represent the convolution between the input and f evaluated at time instant t_i. Let also $\mathcal{K}$ be the kernel on $\mathcal{X} \times \mathcal{X}$ that encodes the expected features of f. Then, we search for the estimate of f in the associated RKHS $\mathcal{H}$ using the squared norm as regularizer, hence generalizing the penalty $\theta^T P^{-1}\theta$ in (32). The resulting estimator is called kernel ridge regression, or also regularization network [119], and turns out to be

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{n} (y_i - L_i[f])^2 + \gamma \|f\|_{\mathcal{H}}^2. \tag{47}$$

This would seem an intricate variational problem possibly defined over an infinite-dimensional space $\mathcal{H}$. Instead, the *representer theorem* says that the solution to (47) is unique and it is computed as the sum of n basis functions. These basis functions are defined by the kernel and the operator L_i and they are scaled with coefficients obtained solving a simple set of linear equations [73], [75], [120]. More details on this will be given during the discussion on nonlinear system identification in the following sections.

RKHSs and Gaussian regression: some historical notes

Theory of RKHSs can be found in the fundamental works [19], [101]. In their simplest formulation they are formulated as Hilbert spaces where any pointwise evaluation of a function is a linear and bounded functional. These spaces are in one-to-one correspondence with positive definite kernels which are called reproducing since, for any function f and any input location, the inner product between $K(x,\cdot)$ and f returns $f(x)$, i.e. $\langle K(x,\cdot), f\rangle_{\mathcal{H}} = f(x)$.

Regularization theory traces back to Tikhonov [102], [103]. First applications of RKHSs can be found in the eighties regarding statistics, approximation theory and computer vision [104], [105], [73], They were then introduced in machine learning by Girosi in [106]. Combination of RKHSs and regularization theory enabled the introduction of algorithms like regularization networks/kernel-ridge regression and support vector machines [107], [17], [108]. In system identification, the concept of stable RKHSs for impulse response and predictor estimation was introduced in [21]. For further developments, see e.g. also [92], [83], [35], [20], [37]. As also described in the book [29], the estimators there introduced have also connections with previous works on regularization in system identification including distributed lag estimators, smoothness regularizers for transfer functions, stochastic embedding [109], [110], [111], [112], [113].

The link between regularization in RKHSs and Bayesian estimation of Gaussian processes was first described in the context of smoothing splines in [24] where one can find the link between Sobolev spaces and integrated white Gaussian noises. An important incentive to the use of Gaussian regression was due to [114] where Neal considered a neural network with weights modeled as i.i.d. random parameters. Using the central limit theorem, he showed that the input-output map becomes a Gaussian process as the network's width grows to infinity. Thus, the function estimate becomes available in closed-form and simplifies to manipulation of covariances. See also [115] and [22]. Recent works generalize [114] by induction and builds new (compositional) kernels as limits of deep neural networks with increasing layer width [116]. These constructions regard also convolutional neural networks [117] and recurrent neural networks [118].

FROM CLASSICAL TO KERNEL-BASED NONLINEAR SYSTEM IDENTIFICATION

The literature on nonlinear system identification is vast and the reasons are manifold. Nonlinearities arise in several engineering problems, e.g. mechanical engineering, robotics, telecommunications, biology and epidemiology [121], [122], [123]. As a consequence, many parameterizations of the unknown system have been introduced along with different estimation methods. Different parameterizations permit to introduce different forms of prior knowledge about the system, leading to *grey box models* with different shades of grey, e.g. see the section *The Palette of Nonlinear Models* in [123]. Our aim here is to give an overview of some aspects of this problem assuming that no prior physical knowledge on the system is available. Hence, the building of a *black-box model* is needed. State-space models estimation will be also discussed.

Classical nonlinear system identification

In the nonlinear context, the ARX model reported in (39) can be generalized as follows. First, we can introduce a vector that contains past input-output data that is

$$x_i = [y_{i-1}, u_{i-1}, \ldots, y_{i-m}, u_{i-m}], \tag{48}$$

where m is the system memory. Then, our model for the noisy output data becomes

$$y_i = f(x_i;\theta) + e_i, \quad i = 1, \ldots, n \tag{49}$$

where the unknown (nonlinear) function f depends on the vector θ. This defines a nonlinear ARX (NARX) model of memory m. A nonlinear FIR (NFIR) is obtained if x_i contains only past inputs, i.e.

$$x_i = [u_{i-1}\ u_{i-2}\ \ldots\ u_{i-m}]. \tag{50}$$

Parametric models linear in θ are widely used. First, d basis functions (the regressors), denoted by ρ_k, are introduced. Then, (49) is rewritten as

$$y_i = \sum_{k=1}^{d} \theta_k \rho_k(x_i) + e_i, \quad i = 1, \ldots, n. \tag{51}$$

Poor knowledge on system dynamics often requires to introduce a large number of basis functions to account for different inputs-outputs lags and interactions. Hence, in such classical framework, nonlinear system identification is interpreted as an extended parametric regression. To control complexity, it is essential to choose the relevant model components, a problem known as regressor selection. This step is key due also to the so called curse-of-dimensionality affecting nonlinear system identification. It can be described via the following simple example. An important model for f that can approximate arbitrarily well any "reasonable" system is the (already mentioned) Volterra series [124], [39]. It corresponds to Taylor expansions of the input-output map in discrete-time. In particular, an NFIR model of order m obtained by an r-truncated Volterra model introduces all the monomials up to degree r. If $r = 2$, the basis functions ρ_k in (51) become

$$\{1, u_{t_i-1}, \ldots, u_{t_i-m}, u^2_{t_i-1}, \ldots, u^2_{t_i-m}, u_{t_i-1}u_{t_i-2}, u_{t_i-1}u_{t_i-3}, \ldots\}, \tag{52}$$

Kernel-based ranking of impulse responses: not all the regularizers are the same

We have explained that the use of a kernel (matrix) P can be seen as a way to introduce a ranking of possible solutions: among impulse responses that fit the data in a similar way, the simplest one (according to the penalty induced by P) has to be chosen. This will be illustrated through an example which will also show how different kernels can be more or less useful for linear system identification.

Let us assume that the impulse response has to be chosen among a finite number of candidates θ, each representing a FIR of length 100. The truth is one of the candidates, visible in the left panel of Fig. 7 (red line, obtained by random generation of a rational transfer function of order 10). The middle panel displays the known input given by filtered white Gaussian noise. The other 99 candidates were fabricated for illustrative purposes as follows. The parameter vector θ of each candidate was obtained by computing (21) with output data defined as the true system output perturbed with a very small noise. The 100 candidates are shown in the left panel: they appear quite different from each other but their convolutions with the input (right panel) reveal that there is no real difference in terms of output fit. This also points out the severe ill-conditioning affecting the problem.

To select the impulse response, some prior information is needed. To this purpose we introduce a kernel-based ranking of impulse responses by measuring FIR complexity through P. Specifically, the impulse response θ^i precedes θ^j if it is assigned a smaller penalty by the kernel, i.e.

$$\theta^i \prec \theta^j \quad \iff \quad (\theta^i)^T P^{-1} \theta^i \prec (\theta^j)^T P^{-1} \theta^j.$$

Hence, P can be seen as a referee that ranks the candidates. First, let us consider the ridge penalty which corresponds to using the identity matrix $P = I_{100}$. The top and middle left panels of Fig. 8 report the two highest-rank impulse responses (black line). One can see that ridge tends to select impulse responses quite far from truth and containing many oscillations. According to the Bayesian interpretation of regularization described before, these are the two candidates that (in some sense) are most similar to white noise realizations. One can then assess that the truth is only in 87th position.

As a second example, let us use the Gaussian kernel, one of the most used models in machine learning to include information on function smoothness. The (i,j)-entry of the kernel is

$$P_{ij} = e^{-\frac{(i-j)^2}{\omega}}, \quad \omega > 0.$$

To remove the dependence on the kernel width ω, the penalty assigned to a generic θ is defined by

$$\min_{\omega} \theta^T P^{-1} \theta.$$

The first and second selected candidates, reported in the first two middle panels, are more regular than those chosen by ridge. In fact, the Bayesian interpretation of regularization reveals that the Gaussian kernel selects vectors most similar to realizations from a stationary process with correlated samples. However, the situation does not improve so much. The true impulse response is in 26th position.

As third example, consider the TC (first-order stable spline) kernel, obtained by setting

$$P_{ij} = \alpha^{\max(i,j)}, \quad 0 \leq \alpha < 1.$$

Similarly to what done in the Gaussian kernel case, the dependence on the hyperparameter is removed by optimizing w.r.t. α, i.e. for any candidate θ we compute

$$\min_{\alpha} \theta^T P^{-1} \theta.$$

The right panels of Fig. 8 show that the true impulse response is now given the first place. The second impulse response is also close to truth. The selected candidates are smooth and decay exponentially to zero, pointing out the importance of the choice of the regularizer for impulse response estimation.

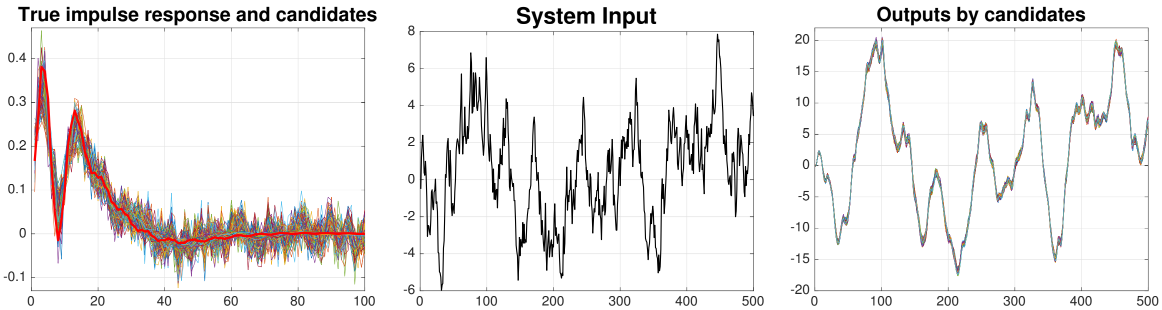

Figure 7 *Kernel-based ranking. Left* System impulse response candidates, including also the truth (red line). *Middle* System input given by a low-pass filtered white Gaussian noise. *Right* Outputs generated by the candidates, i.e. convolution of the impulse responses reported in the top panel with the input in the bottom panel.

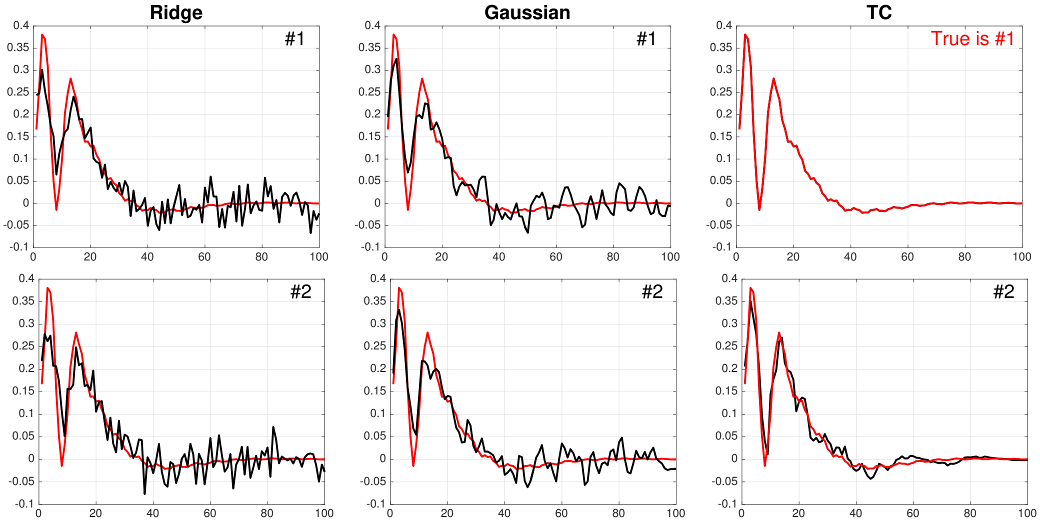

Figure 8 *Kernel-based ranking* Ranking of impulse response candidates using three different kernels: identity (left), associated to ridge regression, Gaussian (middle) and TC/first-order stable spline (right).

and it is easy to see that the model order d, i.e. the overall number of monomials, is given by the binomial coefficient $\binom{m+r}{r}$. In Fig. 9, model order is plotted as a function of m with a small degree r equal to 3. The number of required basis functions scales exponentially with the system memory, outlining how control of model complexity is really an issue.

The previous example thus shows that regressor selection has a combinatorial nature. For this reason, suboptimal solutions are often searched e.g. through greedy approaches like forward orthogonal least squares [125], [126]. and its many variants decribed e.g. in [127][Section 3]. Another approach uses variance analysis (ANOVA) [128] and divide-and-conquer methods [129]. Other strategies include projection pursuit [130] and manifold learning for dimensionality reduction [131], [132], [133].

Use of regularization

An alternative approach is to use regularization adopting sparse promoting penalties. This allows to jointly perform estimation and variable selection, trying to automatically set to zero groups of variables in the regression vector. The use of the ℓ_1-norm as regularizer on θ leads to the famous LASSO [134] and LARS [135]. More recent variants include [136], [137], [138]. However, in [25] it has been shown that the ℓ_1-norm can lead to unsatisfactory results in system identification, even in the linear scenario. LASSO is not so effective to balance bias and variance in dynamic systems, being also much sensitive to the initial choice d of the model dimension. This also holds for other recent regularized approaches for system identification based on atomic and Hankel nuclear norms [139], [140], [141], [142].

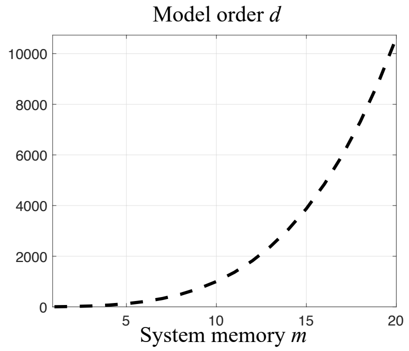

Figure 9 *Curse of dimensionality in nonlinear system identification* Number of basis functions, i.e. monomials, contained in a truncated Volterra (polynomial) model of degree 3. The result is function of the system memory m assuming that the unknown system to reconstruct is $f(x_i)$ where any input location contains m past inputs, i.e. $x_i = [u_{i-1} \dots u_{i-m}]$.

Kernel-based nonlinear system identification

Let us assume that the nonlinear system f belongs to a RKHS $\mathcal{H}$. The identification data are the n couples $\{x_i, y_i\}$, with the input location e.g. given by (48) in the NARX case. According to (49), direct noisy output data of the input-output map are available so that in (47) the term $L_i[f]$ corresponds to $f(x_i)$. Our regularized NARX (or NFIR) estimator is thus given by

$$\hat{f} = \arg\min_{f\in\mathcal{H}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \gamma\|f\|^2_{\mathcal{H}}. \qquad (53)$$

The application of the representer theorem cited at the end of the previous section, permits to obtain $\hat{f}$ in closed form. Let $Y = [y_1, \dots, y_n]^T$ while $\mathbf{K}$ is the so called kernel matrix whose dimension is $n \times n$ with (i,j)-entry $\mathbf{K}_{ij} = \mathcal{K}(x_i, x_j)$. The estimate of the nonlinear system has the structure of a particular neural network with only one layer where the weights $\hat{c}_i$ solve a linear system of equations. Specifically, one has

$$\hat{f}(x) = \sum_{i=1}^{n} \hat{c}_i K_{x_i}(x) \quad \forall x \tag{54}$$

with the weights vector given by

$$\hat{c} = (\mathbf{K} + \gamma I_n)^{-1} Y, \tag{55}$$

with I_n the $n \times n$ identity matrix.

From a computational viewpoint, the main drawback to compute $\hat{f}$ is the inversion of the matrix $\mathbf{K} + \gamma I_n$ whose computational cost is $O(n^3)$. This problem has been also connecting machine learning and convex optimization [143], [144], [145]. Numerical techniques include approximate representations of the kernel function [146], [147] based on the Nyström method or greedy strategies [148], [149], [150]. Low-order kernel approximations are also employed by truncating the expansion (46), see [151], [152], [153], [154], [155]. Other randomized approaches are described in [156], [157], [45], [47]. See also [158] for a recent survey.

Bayesian interpretation: Gaussian regression of random fields

In the linear setting we have seen that the kernel-based impulse response estimator can be seen as a minimum variance estimator if θ is a zero-mean normal vector of covariance proportional to the kernel, see (41). A similar relationship holds also in the nonlinear setting with the impulse response replaced by the input-output relationship f seen as a nonlinear random surface, see e.g., [22]. Specifically, we model f as a zero-mean Gaussian random field, i.e. given any finite collection of input locations $\{x_i^*\}_{i=1}^p$, the sampled function $[f(x_1^*) \; \dots \; f(x_p^*)]$ forms a Gaussian vector. Let the covariance of such vector be $\lambda\mathbf{K}$, where λ is a positive scale factor and $\mathbf{K}$ is the kernel matrix with (i,j)-entry $\mathbf{K}_{ij} = \mathcal{K}(x_i^*, x_j^*)$. Our system outputs are

$$y_i = f(x_i) + e_i, \quad i = 1, \dots, n$$

where, as usual, the noise is white and Gaussian, of variance σ^2 and independent of f. Now, using basic results on estimation of jointly Gaussian vectors [76], one can still assess that the posterior mean $\mathcal{E}(f(x)|Y)$ coincides with (54) and, in turn, with (53) just setting $\gamma = \frac{\sigma^2}{\lambda}$. This interpretation is also useful for hyperparameters tuning. Letting $Z(\eta) = \lambda\mathbf{K}(\eta) + \sigma^2 I_n$, the marginal likelihood estimate of η has the same expression obtained in the linear setting in (44), i.e.

$$\hat{\eta} = \arg\min_{\eta} \; Y^T \left(Z(\eta)\right)^{-1} Y + \log\det(Z(\eta)). \tag{56}$$

Kernels for nonlinear system identification

In the nonlinear setting, the kernel has to describe the nonlinear input-output relationship. The associated RKHS contains functions over $\mathcal{X}$ with dimension related to the system memory m.

Kernels can face the curse of dimensionality by the implicit encoding of functions described by (46). A relevant example is the polynomial kernel

$$\mathcal{K}(x,z) = \left(\langle x, z\rangle_2 + 1\right)^r, \quad r \in \mathbf{N}, \tag{57}$$

where $\langle \cdot, \cdot \rangle_2$ indicates the classical Euclidean inner product. In the NFIR case, it relates to the truncated Volterra series since (57) embeds all the monomials up to the r-th degree. Hence, one has $d = \binom{m+r}{r}$. From (54) and (55), one can see that monomials' encoding has important computational advantages since estimation complexity, even if cubic in the number n of output data, is linear in the system memory m and independent of the degree r of nonlinearity. Even if the polynomial kernel may induce a very rich class of functions depending on the degree r, the expansion (46) contains a finite number d of monomials. Hence, the RKHS induced by (57) is always finite-dimensional. It is also possible to use universal kernels that can approximate any continuous function [159]. The most notable example is the Gaussian kernel. It was previously defined over the set of natural numbers in (33) as a possible description of an impulse response. In the nonlinear setting, it is defined over a multi-dimensional domain as follows

$$\mathcal{K}(x,a) = e^{\frac{-\|x-a\|^2}{\omega}} \tag{58}$$

where ω is still the kernel width and $\|\cdot\|$ is the classical Euclidean norm. The Gaussian kernel is widely used to describe input-output relationships just known to be smooth. According to the Bayesian interpretation, where the kernel is proportional to the covariance, it is associated with a stationary random field. Fig. 10 (left) plots a realization that gives an idea of the model underlying the Gaussian kernel in the NFIR case, with system memory $m = 2$.

Enriching the Gaussian kernel with a non stationary component can be useful in many circumstances, e.g. to model linear components present in the dynamic system. This point is related to the literature on partial linear models [160], [161], [162]. One simple approach is to add to the Gaussian kernel a linear kernel, hence obtaining

$$\mathcal{R}(x_i, x_j) = \lambda_L x_i^T P x_j + \lambda_{NL}\mathcal{K}(x_i, x_j) \tag{59}$$

with the matrix P defined e.g. by the stable spline kernel (36). The two scale factors λ_L and λ_{NL} balance the relative power of the linear and nonlinear system part and can be tuned by marginal likelihood optimization. A realization from the Gaussian random field which includes the linear part is in the right panel of Fig. 10.

The use of kernels like (59) may require the tuning of the system memory forcing to use grids. This problem can be circumvented by following stable spline-like ideas, incorporating fading memory concepts in the classical Gaussian kernel. One can include the information that u_{i-k} is expected to have less influence on y_i as the lag k increases. Considering just for simplicity the NFIR case, this can be obtained by redefining $\mathcal{K}$ in (59) as follows

$$\mathcal{K}(x_i,x_j)=\exp\Big(-\sum_{k=1}^{m}\alpha_{NL}^{k-1}\frac{(u_{i-k}-u_{j-k})^2}{\omega}\Big),\quad 0<\alpha_{NL}\leq 1. \tag{60}$$

This model is known as nonlinear stable spline kernel in the literature [163]. The hyperparameter α_{NL} is to model the exponential decay of the influence of past inputs' on the output. Hence, one can set m to a large value. Then, the decay hyperparameters α and α_{NL} present in P and $\mathcal{K}$, respectively, will decide the effective memory of the linear and nonlinear system's part.

Kernel-based estimation of state-space models

In many cases, physical phenomena can be more easily modeled by state-space descriptions e.g. given by

$$x_{i+1}=\mathbf{f}(x_i)+\mathbf{e}_i,\quad i=1,\dots,n \tag{61}$$

where x_i is the d-dimensional state at instant i while $\mathbf{e_i}$ contains the d random noises. The function $\mathbf{f}$ is vector-valued and encapsulates d transition functions which we denote with f_k for $k=1,\dots,d$. If the system states are all observable, each transition function can be estimated by (53) just noting that the states x_i define both input locations and measurements. Specifically, if y_{ik} indicates the k-th component of x_{i+1}, our kernel-based estimators of the f_k are

$$\hat{f}_k=\arg\min_{f\in\mathcal{H}}\sum_{i=1}^{n}(y_{ik}-f_k(x_i))^2+\gamma\|f\|_{\mathcal{H}}^2,\ \ k=1,\dots,d. \tag{62}$$

A closed form expression for $\hat{f}_k$ can be obtained using formulas (54) and (55).

This stochastic view can be exploited also to identify more complex state space models, in particular when some of the states are not measurable. In this context, equation (61) is coupled with the output equation

$$y_i=\mathbf{g}(x_i)+\mathbf{v}_i$$

where also the function $\mathbf{g}$ needs to be estimated.

In [164], the state transition function $\mathbf{f}$ is modeled as a GP while $\mathbf{g}$ as a parametric likelihood of the form $p(y_i|x_i,\theta_y)$, that is, the observation model $\mathbf{g}$ is assumed to be parameterized by the finite dimensional parameter θ_y. Through sophisticated estimation tools, the authors developed strategies to estimate the posterior $p(\mathbf{x}_{[0,T]}|\mathbf{y}_{[0,T]})$, where $\mathbf{y}_{[0,T]}$, $\mathbf{x}_{[0,T]}$ denote, respectively, the time-series of measurements and states from time 0 up to T. Inferring the distribution over the state trajectory $p(\mathbf{x}_{[0,T]}|\mathbf{y}_{[0,T]})$ is an important problem in itself known as smoothing. In particular in [164] a tailored particle Markov Chain Monte Carlo (PMCMC) algorithm is used to efficiently sample from the smoothing distribution whilst marginalizing over the state transition function. Once an approximation of the smoothing distribution is obtained, with the dynamics of the model marginalized out, learning the function $\mathbf{f}$ is straightforward since its posterior is available in closed form given the state trajectory.

BOUNDS FOR SYSTEM IDENTIFICATION - PART A: THE BAYESIAN & GAUSSIAN SET-UP

This section and the next one address the problem of quantifying the uncertainty about an identified system. "Part A" complements the exposition so far with useful error bounds that are derived based on the Bayesian interpretation of kernels, for fixed values of the hyperparameters. These bounds are meaningful in a Gaussian-Bayesian framework, where the user attaches a (Gaussian) probability not only to the noise (which is the typical starting point in statistical system identification) but also to the possible candidate system models. This framework is elegant and effective: a simple inference rule (the Bayes' rule) leads to rigorous conclusions, by means of computations that are eased by the Gaussian assumption. Nonetheless, the sensitivity of the bounds to the working assumptions and, specifically, to the postulated probability distributions, may be a legitimate source of concern, encouraging both the scientist and the user to step back, at least for a moment, from the Bayesian-Gaussian framework, and to look for alternative and complementary points of view. In "Part B", the interested reader can find a brief overview of alternative approaches to the computation of error bounds, including some that are rooted in the tradition of system identification and some that are the subject of active, and challenging, research efforts; "Part B" can be skipped at first reading without loss of continuity.

Bounds for linear systems

Systems in linear regression form:

The linear regression problem (20) has been formulated as a Gaussian regression problem in "Bayesian interpretation: Gaussian regression", where Φ is treated as a fixed quantity, while the noise and θ are independent Gaussian; consequently, the joint probability distribution of θ and Y is Gaussian, namely:

$$\begin{bmatrix}\theta\\ Y\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}\mathcal{E}(\theta)\\ \mathcal{E}(Y)\end{bmatrix},\begin{bmatrix}\mathrm{Var}(\theta) & \mathrm{Cov}(\theta,Y)\\ \mathrm{Cov}(Y,\theta) & \mathrm{Var}(Y)\end{bmatrix}\right), \tag{63}$$

with $\mathcal{E}(\theta)=0$, $\mathcal{E}(Y)=0$, $\mathrm{Var}(\theta)=\mathcal{E}(\theta\theta^T)=\lambda P$, $\mathrm{Var}(Y)=(\lambda\Phi P\Phi^T+\sigma^2 I_n)$, $\mathrm{Cov}(\theta,Y)=\lambda P\Phi^T$. The *posterior* distribution, which is denoted by $p(\theta|Y)$, is the conditional distribution of θ given Y and is a complete descriptor of the remaining user's uncertainty about the latent θ after that θ has (partially) revealed itself through the observed data Y. The posterior distribution is obtained from (63) by the Bayes' rule, and turns out to be Gaussian

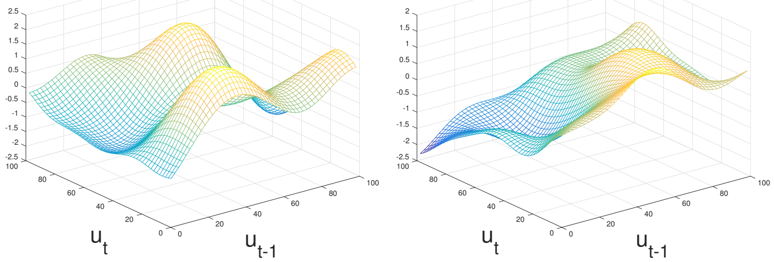

Figure 10 *Bayesian interpretation of regularization in the nonlinear setting* Realization from a zero-mean random Gaussian field with covariance equal to the Gaussian kernel (left) and to the sum of a Gaussian kernel and a linear kernel (right). In this latter case a linear trend is generated to describe the linear part of the system.

as well:

$$p(\theta|Y) = \mathcal{N}(\mathcal{E}(\theta|Y), \mathrm{Var}(\theta|Y)),$$

where $\mathcal{E}(\theta|Y)$ is the conditional expectation and $\mathrm{Var}(\theta|Y)$ the conditional variance; $\mathcal{E}(\theta|Y)$ and $\mathrm{Var}(\theta|Y)$, which together specify completely the posterior, can be explicitly computed by using well-known identities for Gaussian distributions, i.e.,

$$\mathcal{E}(\theta|Y) = \mathrm{Cov}(\theta, Y)(\mathrm{Var}(Y))^{-1}Y,$$

$$\mathrm{Var}(\theta|Y) = \mathrm{Var}(\theta) - \mathrm{Cov}(\theta, Y)(\mathrm{Var}(Y))^{-1}\mathrm{Cov}(Y, \theta),$$

which, by simple substitutions, lead to expressing $\mathcal{E}(\theta|Y)$ as in (32) with $\gamma = \frac{\sigma^2}{\lambda}$ (thus, as already noticed, $\mathcal{E}(\theta|Y)$ coincides with the kernel estimate $\hat{\theta}$ when $\gamma = \frac{\sigma^2}{\lambda}$), and to

$$\begin{aligned}\mathrm{Var}(\theta|Y) =& \lambda P - \lambda P\Phi^T(\lambda\Phi P\Phi^T + \sigma^2 I_n)^{-1}\Phi\lambda P; \text{or} \\ =& \left(\frac{1}{\sigma^2}\Phi^T\Phi + \frac{1}{\lambda}P^{-1}\right)^{-1}. \end{aligned} \quad (64)$$

The distribution $p(\theta|Y)$ can then be used at will to evaluate the uncertainty about θ after observing Y. It is rather natural to summarize the uncertainty by means of regions around the kernel estimate $\hat{\theta}$. For example, the uncertainty on the component θ_i is fully described by the marginal Gaussian distribution with mean $\hat{\theta}_i = [\mathcal{E}(\theta|Y)]_i$ (i-th component of $\mathcal{E}(\theta|Y)$) and standard deviation equal to $\sigma_i = \sqrt{[\mathrm{Var}(\theta|Y)]_{i,i}}$ (square root of the i-th diagonal element of the matrix $\mathrm{Var}(\theta|Y)$), and a 95% probability interval for θ_i can then be computed as

$$[\hat{\theta}_i - 1.96\sigma_i, \hat{\theta}_i + 1.96\sigma_i].$$

Such an interval is said to be a *credible interval* at level $1-\epsilon$, with $\epsilon = 5\%$. Similarly, a credible region Θ_{Bayes} at level $1-\epsilon$ for the whole θ can be constructed by considering the smallest volume d-dimensional region that has probability $1-\epsilon$ according to $p(\theta|Y)$. Such a minumum volume region is an ellipsoid, centred at $\hat{\theta} = \mathcal{E}(\theta|Y)$, that can be written as $\Theta_{Bayes} =$

$$\left\{\theta : (\theta - \hat{\theta})^T \left(\frac{1}{\sigma^2}\Phi^T\Phi + \frac{1}{\lambda}P^{-1}\right)(\theta - \hat{\theta}) \leq F_{\chi^2_d}(1-\epsilon)\right\},$$

where $F_{\chi^2_d}(\cdot)$ is the cumulative distribution function of the chi-squared distribution with $d = \dim(\theta)$ degrees of freedom.

Systems in linear regression form with autoregressive part:

In the set-up of (39)&(40), the formulas (32)&(64) turn out to be still valid, see e.g. [92] for a complete study. This fact can be illustrated on the simple autoregressive system

$$y_i = \theta y_{i-1} + e_i, \quad (65)$$

where θ and the noise are independent Gaussian, and the initial condition y_0 is given. First, let us assume that only y_1 has been observed. With $\Phi = y_0$ and $Y = [y_1]$, θ and Y are jointly Gaussian for any fixed value of Φ, so the reasoning leading to (63) applies verbatim. Let us assume, instead, that the observations are y_1, y_2; then, the components of the regressor $\Phi = [y_0, y_1, y_2]$ and of the observation vector $Y = [y_1, y_2]$ overlap, which calls for extra care. Nonetheless, the sought distribution $p(\theta|Y)$ is as in the non-autoregressive case: in fact, $p(\theta|y_2, y_1, y_0)$ can be factored as $p(\theta|y_2, y_1, y_0) \propto p(y_2|y_1, \theta)p(y_1|y_0, \theta)p(\theta)$ (we used that y_0 is independent of θ).

Linear systems in state-space form:

The uncertainty on the identified system matrices can be fully described by means of the posterior $p(\theta|Y)$ of the identified parameters, regardless of whether the matrices are obtained directly by least squares or by computing a state-state space realization of the identified input-output transfer function, as discussed in "Linear state-space models". Computing uncertainty regions for the unknown matrix elements is just one of the many possible usages of the posterior $p(\theta|Y)$, which is useful to target a multitude of problems. When these problems require complex transformation of the identified parameters, a precious ally for the actual computation of the results is Monte Carlo sampling. Two examples follow: (example 1) in order to evaluate the probability that an unknown system is unstable, it is sufficient to sample many independent instances of the identified parameter vector according to $p(\theta|Y)$, use these samples to build many instances of the unknown system matrix F, say $\hat{F}^{(1)}, \hat{F}^{(2)}, \dots, \hat{F}^{(m)}$, and count how many of these matrices have unstable eigenvalues; (example 2) a control policy can be chosen as the one that performs better on a sample of systems obtained according to $p(\theta|Y)$ (see e.g. [165], [57], [166]).

Bounds for nonlinear system identification

The formulas for linear systems can be generalized to the nonlinear case. First, let us consider the set-up already discussed in "Bayesian interpretation: Gaussian regression of random fields"

$$y_i = f(x_i) + e_i, \tag{66}$$

where the input values x_i are fixed, f is modeled as a Gaussian random field, and e_i is an independent Gaussian noise. Here, the conditional probability of the random field f given the data $Y = [y_1, \dots, y_n]^T$ is still Gaussian and the uncertainty about f can be evaluated at any sequence of user-chosen inputs $x_1^*, x_2^*, \dots, x_m^*$ by computing the distribution of the random vector $f_{x^*} := [f(x_1^*), f(x_2^*), \dots, f(x_m^*)]^T$, which is (see e.g. [22], Chapter 2)

$$p(f_{x^*}|Y) = \mathcal{N}(\mathcal{E}(f_{x^*}|Y), \mathrm{Var}(f_{x^*}|Y)),$$

with

$$\mathcal{E}(f_{x^*}|Y) = \lambda \mathbf{K}_{x^*,x}(\lambda \mathbf{K}_{x,x} + \sigma^2 I_n)^{-1} Y, \tag{67}$$

where $\mathbf{K}_{x^*,x}$ is the $m \times n$ matrix with (i,j)-entry $\mathcal{K}(x_i^*, x_j)$ and $\mathbf{K}_{x,x}$ the $n \times n$ matrix with (i,j)-entry $\mathcal{K}(x_i, x_j)$, and with

$$\mathrm{Var}(f_{x^*}|Y) = \lambda \mathbf{K}_{x^*,x^*} - \lambda \mathbf{K}_{x^*,x}^T (\lambda \mathbf{K}_{x,x} + \sigma^2 I_n)^{-1} \lambda \mathbf{K}_{x^*,x}. \tag{68}$$

As usual, the formula of the conditional expectation $\mathcal{E}(f_{x^*}|Y)$ coincides with that of the kernel estimate $\hat{f}$ evaluated at the input points $x^* = [x_1^*, x_2^*, \dots, x_m^*]^T$ (provided that $\gamma = \frac{\sigma^2}{\lambda}$). Notably, (67) and (68) generalize the equations (32) and (64), which are recovered when $f(z) = z^T\theta$ and $K(z, z') = z^T(\lambda P)z'$. Then, the uncertainty of f at a given input point x_i^*, i.e., of $f_{x_i^*} = f(x_i^*)$, can be described by the Gaussian posterior with mean $\hat{f}(x_i^*) = \lambda \mathbf{K}_{x_i^*,x}(\lambda \mathbf{K}_{x,x} + \sigma^2 I_n)^{-1} Y$ and conditional standard deviation $\sigma_{x_i^*} = \sqrt{\lambda K(x_i^*, x_i^*) - \lambda \mathbf{K}_{x_i^*,x}^T (\lambda \mathbf{K}_{x,x} + \sigma^2 I_n)^{-1} \lambda \mathbf{K}_{x_i^*,x}}$. The corresponding credible interval at level 95% for $f(x_i^*)$ is

$$[\hat{f}(x_i^*) - 1.96\sigma_{x_i^*}, \hat{f}(x_i^*) + 1.96\sigma_{x_i^*}].$$

Analogously, the minimum-volume credible region of probability $1 - \epsilon$ for f_{x^*} is the ellipsoid

$$\left\{ f_{x^*} : (f_{x^*} - \hat{f}_{x^*})^T C_{x^*}^{-1} (f_{x^*} - \hat{f}_{x^*}) \leq F_{\chi_m^2}(1 - \epsilon) \right\},$$

where $\hat{f}_{x*} = \mathcal{E}(f_{x^*}|Y)$ and C_{x^*} is short for $\mathrm{Var}(f_{x^*}|Y)$.

In nonlinear system identification, x_i can be defined as a vector of system inputs, as in (50). Importantly, like in the linear case, the same formulas (67)&(68) apply to systems with an autoregressive part, that is, to situations where x_i includes both system input and output values, as in (48) or in *state-space system identification*. In this latter case, valid bounds are obtained by establishing a mapping between the terms in (66) and the relevant state-space variables according to the discussion in "Kernel-based estimation of state-space models", i.e., as follows

» the input x_i is the system state at time i,

» the unknown function f is the k-th component of the system transition function,

» the observation y_i is the k-th component of the system state at time $i + 1$.

This forms the basis of the estimation part of the algorithm PILCO [65], which will be discussed in more detail in the section "Model-Based Reinforcement Learning". A generalization to multi-step prediction is available in [167], while the case of states that are not directly observable is studied in [164].

BOUNDS FOR SYSTEM IDENTIFICATION - PART B: BEYOND THE BAYESIAN & GAUSSIAN SET-UP

In this part, approaches outside the Bayesian & Gaussian set-up are considered. For the sake of space constraints, the discussion will be focused on estimating the impulse response vector g in (11), and limited to a subset of techniques (e.g., system identification in the frequency domain, [168], will not be discussed). Nonetheless, many of the concepts here revisited are general enough to be applicable to more complex linear or nonlinear systems such as (39) and (49) (in fact, more general systems than (11) are typically addressed in the literature that will be referenced throughout this section). To begin with, it is assumed that d, the length of g, be known and sufficiently small. At least three approaches for the construction of bounds on the estimation error can then be distinguished based on the mathematical description of the uncertainty.

I) Uncertainty as a set of possibilities

If the noise sequence $(e_1, e_2, \dots, e_n)$ is known to belong to a set of possible sequences E_{poss}, then the observed input-output sequence $\{(u_i, y_i)\}$ can be used, together with a candidate impulse response $\{\tilde{g}_k\}_{k=1}^d$, to compute the residuals $\tilde{e}_i = y_i - \sum_{k=1}^d u_{i-k}\tilde{g}_k$, $i = 1, \dots, n$, which coincide with the actual noise variables $e_1, e_2, \dots, e_n$ when $g = \tilde{g}$. Thus, given certain input-output measurements, we say that $\tilde{g}$ belongs to the set of compatible impulse responses, which we denote by Θ_{poss}, if and only if $(\tilde{e}_1, \tilde{e}_2, \dots, \tilde{e}_n) \in E_{poss}$. This idea is at the core of the vast literature on *set membership* system identification, see e.g. [169], [170], [171], [172], [173], [174], [51], [175], [176], [177]. If E_{poss} is correctly specified, the true g certainly belongs to the uncertainty set Θ_{poss}. However, the definition of E_{poss} is critical: if all the imaginable noise realizations are included in E_{poss}, the set Θ_{poss} ends up being conservative and uninformative for practical purposes. On the other hand, removing some noise realizations from the set E_{poss} typically invalidates the claim that Θ_{poss} *certainly* includes g. In engineering, taking risks is often acceptable, provided that they are quantified and suitably weighed. This leads to the probabilistic approach.

II) Uncertainty as probability

A natural way to account for the risk due to neglecting a subset of E_{poss} is introducing a probability measure $\mathbb{P}$ over E_{poss}. In fact, *introducing* $\mathbb{P}$ *makes it possible to quantify how likely and important are in our eyes and for our purposes certain subsets of realizations*. Along with the probabilistic approach, a subset $E_{neglected}$ of E_{poss} with small probability ϵ (i.e., such that $\mathbb{P}(E_{neglected}) = \epsilon$, where ϵ is, for example, 0.01) can be isolated. Then, *given a set of input-output data*, a set of candidate impulse responses Θ_{prob} is obtained according to the following definition.

> DEFINITION: Θ_{prob} is the set of the impulse responses $\{\tilde{g}\}_{k=1}^{d}$ for which the residuals $\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_n$ (with $\tilde{e}_i = y_i - \sum_{k=1}^{d} u_{i-k}\tilde{g}_k$) belong to the set $E_{poss} \setminus E_{neglected}$ (of probability $1-\epsilon$).
>
> (69)

It is *possible* that the set Θ_{prob} constructed based on the observed data does not include the true g. Nonetheless, by the very definition of Θ_{prob}, the user can make the following claim: "$g \in \Theta_{prob}$ *unless* an unlikely event of probability ϵ has happened". In a common terminology, $1-\epsilon$ is a *confidence value*, and Θ_{prob} is said to be a *confidence region* at level $1-\epsilon$ for g (as is clear from the definition, the terms *confidence regions* and *credible regions* denote different concepts; for more details see also the sidebar "A Bayesian-frequentist interpretation of the bounds"). Many elaborations are possible; for example, $E_{neglected}$ can be defined in such a way that Θ_{prob} is maximally informative, according to some optimality criteria, see e.g. [178].

In the probabilistic approach, the criticality of defining E_{poss} gives way to the criticality of defining $\mathbb{P}$, which can be an even more difficult task. This difficulty explains why classic system identification methods do not usually postulate that a complete probabilistic description $\mathbb{P}$ of the noise process is available (an exception being the archetypical Gaussian noise set-up), but only that suitable technical conditions for the applicability of the central limit theorem are satisfied. While the reader is referred to classic textbooks such as [2] and [3] for details, the basic idea in the least squares set-up is that, as the number of data n tends to infinity, the re-scaled estimation error vector $\sqrt{n}(g - \hat{g})$ tends to have a Gaussian distribution with zero mean and finite, known covariance, no matter what the specific distribution $\mathbb{P}$ is. Thus, a classical way to describe the uncertainty of the least squares estimate θ^* of g (21), uses this asymptotic Gaussian distribution to shape an uncertainty ellipsoid Θ_{Gauss} around θ^*:

$$\Theta_{Gauss} = \left\{ \theta : (\theta - \theta^*)^T \Psi (\theta - \theta^*) \leq \frac{1}{n}\sigma^2 F_{\chi^2_d}(1-\epsilon) \right\}, \tag{70}$$

where $\Psi = \lim_{n\to\infty} \frac{1}{n}\Phi^T\Phi$, and σ^2 is the variance of the stationary, zero-mean noise process $\{e_k\}_{k=1}^{+\infty}$. It is common practice to replace, Ψ and σ^2 with their finite-sample estimates $\hat{\Psi}_n = \frac{1}{n}\Phi^T\Phi$ and $\hat{\sigma}_n^2 = \frac{1}{n-d}\|Y - \theta^*\Phi\|^2$: interestingly, since $\hat{\sigma}_n^2$ tends to σ^2 as $n \to \infty$, the user does not really need to know σ^2, which can rather be considered as an estimable parameter of the unknown distribution $\mathbb{P}$.

This classic way of proceeding is attractive because it is simple and bears a promise of wide applicability and objectivity, in the following precise sense: two users, Alice and Bob, who observe the same input-output data but postulate two different probability distributions for the stationary noise (say $\mathbb{P}_{Alice}$ and $\mathbb{P}_{Bob}$), not only get the same uncertainty ellipsoid, but also agree that, *asymptotically*, Θ_{Gauss} becomes a valid confidence region at level $1-\epsilon$. However, the word "asymptotically" cannot be safely omitted in the previous sentence, and relying on asymptotic results can be deceiving in real life, where only a finite sample of data is available (see e.g. [180], [181]).

In concluding, classic results have two attractive features: they are (i) probabilistic and (ii) robust with respect to a large variety of probabilistic formulations that may be adopted to describe the uncertainty. Unfortunately, they are not valid for finite samples of data.

III) Probabilistic and robust approach

Remarkably, *it is* possible to construct finite-sample valid confidence regions by exploiting only some rather general statistical features of the noise process, a notable example of these features being *stochastic symmetry* (which we call just symmetry).

For example, two independent zero-mean Gaussian noise variables e_1, e_2 are symmetric because, conditioning on their absolute values $|e_1|, |e_2|$, the 4 possible sequences $(+|e_1|, +|e_2|)$, $(+|e_1|, -|e_2|)$, $(-|e_1|, +|e_2|)$, $(-|e_1|, -|e_2|)$ are equally probable. Besides Gaussian, infinitely many other distributions are symmetric. Moreover, for the symmetry property to hold, a noise sequence $e_1, e_2, \ldots, e_n$ need not be identically distributed, and one can easily see that even independence is not necessary. In this sense, symmetry is a mild assumption. Nonetheless, under the symmetry assumption, it is possible to define (i) valid and (ii) informative confidence regions Θ_{prob}:

(i) The possibility to define *valid* regions is implied by the existence of statistical tests for symmetry that are valid at a user-chosen level $1-\epsilon$. A simple example of such a test is the following one: a sequence $\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_{10}$ *passes* the test if and only if $\sum_{i=1}^{10} \operatorname{sign}(\tilde{e}_i) < 10$. Then, a symmetric sequence *fails* the test whenever it is all made of positive

A Bayesian-frequentist interpretation of the bounds

To further remark the difference between a credible region Θ_{Bayes} and a confidence region Θ_{prob}, confidence regions like Θ_{prob} are sometimes called *frequentist* confidence regions. This terminology can be motivated as follows. Let us fix g in system (11) and then consider M repeated thought experiments: in each experiment, data are generated by independently resampling the noise; for each experiment, let us construct the confidence region at level $1-\epsilon$, denoted by $\Theta^{(i)}_{prob}$, computed using the i-th experiment data; let us compute the frequency freq$_M$ with which the so-obtained M regions include the fixed g, i.e., freq$_M = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}\{g \in \Theta^{(i)}_{prob}\}$ ($\mathbf{1}\{g \in \Theta^{(i)}_{prob}\} = 1$, if $g \in \Theta^{(i)}_{prob}$; $\mathbf{1}\{g \in \Theta^{(i)}_{prob}\} = 0$, otherwise); it can be seen that freq$_M$ tends to $1-\epsilon$ as $M \to \infty$. In general, credible regions are not valid frequentist confidence regions: in fact, in an analogue series of mental experiments, the credible regions at level $1-\epsilon$ will include the fixed g with a frequency that departs from $1-\epsilon$ and depends on the specific value of g. The reverse implication is also false: in general, confidence regions at level $1-\epsilon$ are not credible regions at level $1-\epsilon$. However, a common ground for (Bayesian) credible regions and (frequentist) confidence regions can be found: let us consider again M independent experiments but assume that, in each experiment, an instance of the noise sequence *and* an instance $g^{(i)}$ of g are drawn together according to the Bayesian distribution $\mathbb{P}_{g,noise}$ that models the uncertainty with respect to both the noise and the model g. Then, the frequencies $\frac{1}{M}\sum_{i=1}^{M} \mathbf{1}\{g^{(i)} \in \Theta^{(i)}_{prob}\}$ and $\frac{1}{M}\sum_{i=1}^{M} \mathbf{1}\{g^{(i)} \in \Theta^{(i)}_{Bayes}\}$ both converge to $1-\epsilon$. Any region (whether it be a credible region, a confidence region, or none of the above) that includes g with a $1-\epsilon$ rate when samples are repeatedly drawn from the Bayesian prior $\mathbb{P}_{g,noise}$ is said to be a valid *Bayesian-frequentist* region, [179].

values, which happens with probability $(1/2)^{10} = \frac{1}{1024}$ only. Thus, by calling $E_{neglected}$ the subset of E_{poss} where the test fails, a region Θ_{prob} constructed according to the usual definition (69) is necessarily a confidence region at level $1 - \frac{1}{1024}$.

(ii) The possibility to construct regions that are really *informative* for the purpose of system identification is much less obvious. Informally, the more a candidate $\tilde{g}$ differs from the true g, the higher the probability must be that the corresponding residuals $\tilde{e}_1, \dots, \tilde{e}_n$ belong to $E_{neglected}$. The *Sign-Perturbed-Sums* (SPS) algorithm, [182], enforces this property by building a statistical test upon the normal equations encountered in least squares estimation. In this way, testing for symmetry is connected to the system identification goal and the gap between asymptotic optimality and finite-sample validity is filled ([183] contains the proof that, as n tends to infinity, the uncertainty region constructed by SPS becomes more and more similar to the classic region (70)).

For more results about SPS and other methods based on the principles here briefly outlined, the interested reader is referred to the overview [184] and the related works [185], [186], [187], [182], [188], [189], [190], [191], [192], [193], [194], [195]. See also the sidebar "The limits of learning".

The Bayesian approach revisited

As it was amply discussed, the estimation problem becomes quickly ill-posed when d is taken as a large value. While regularization fixes ill-posedness at a technical level (variants of SPS that incorporates a regularization term were proposed in [211], [212], [213], [214]), there are intrinsic limits on the information that a small amount of data can carry about a large amount of parameters. Thus, limitations on the possible model structures are often introduced, [215], while leaving open the possibility to detect undesired undermodelling (see e.g. [216], [217]). A more radical approach prescribes to attach a probability to the set of the infinite many possible candidate system models, similarly to what is normally done with the set of the possible noise realizations in the standard probabilistic approach. This idea is at the core of *Stochastic Embedding*, [218], [219], [220], and has been revitalized by the Bayesian interpretation of kernel methods (see also [221] for the relationship between Stochastic Embedding and kernel methods). In this framework, a probability $\mathbb{P}_{g,noise}$ weights the realizations of both the noise and the unknown g. In the Gaussian set-up, this leads to the bounds in "Part A", while, in the more difficult non-Gaussian set-up, numerical and randomized methods, such as Markov chain Monte Carlo techniques, are particularly helpful to compute the bounds, see e.g. [94], [222].

However, it must always be reminded that two users, Alice and Bob, that use different distributions $\mathbb{P}_{g,noise}$ will end up computing different posterior distributions and, therefore, construct different credible regions Θ_{Bayes}. In particular, they will draw different conclusions about whether their regions are valid credible regions at level $1-\epsilon$ or not.

Beyond Bayesian

A first approach, already employed in this paper, to robustify the Bayesian prior $\mathbb{P}_{g,noise}$ is introducing hyperparameters in the definition of $\mathbb{P}_{g,noise}$: if the different beliefs of Alice and Bob can be reduced to a different choice of a parameter, which we denote by η, then Alice and Bob can step back from their beliefs about η and try to estimate η

The limits of learning

The analysis of finite-sample algorithms offers a constructive way to show that certain systems can be "learnt" from data. In fact, finite-sample algorithms like the Sign-Perturbed-Sums (SPS) algorithm, [182], provide a direct, data-oriented approach to the construction of bounds, which are constructed based on complex manipulations of the available data, so that informative data lead to smaller uncertainty regions while noninformative data lead to larger regions. In particular, in SPS, the shape of the region depends on the outcome of suitable symmetry tests, and some theoretical investigations have been carried out to study under which conditions the ensuing regions are well-shaped, [182], bounded, [196], shrink and at which rate, [183], etc. A complementary approach to the learning problem is offered by the literature that investigates the limits of learning by connecting learning rates to certain characteristics of the system (e.g., its stability radius, the variance or the sub-Gaussian norm of the noise, etc.), see e.g. [197], [198], [199], [200], [201], [202], [203], [204], [205], [206], [207], [87], [208], [209], [210]. Typically, from these studies, finite-sample bounds around $\hat{g}$ can be obtained that depend on certain features of the data generating systems. These bounds have a theoretical value and play a role as a sanity-check for uncertainty regions, like those provided by SPS, which exploit in more complex ways the observed data. Differently from SPS bounds, these bounds are not engineered to produce tight confidence regions at level $1-\epsilon$. Nonetheless, the gap with respect to tight approaches like SPS can be reduced to some extent by estimating some relevant features of the system from data.

from the available data, e.g., by considering the most likely value of η given the observations. This idea is employed in *Empirical Bayes* methods, [53], [54], [52], and generalizes the classic trick of replacing σ with the data-based estimate $\hat{\sigma}_n$ in (70). However, finite-sample error bounds that are valid for both Alice and Bob are hardly obtained in the presence of tunable hyperparameters, unless η is, in turn, treated as a random variable, and Alice and Bob agree on its distribution (in this case, in fact, the inferences of both Alice and Bob can be carried out in a fully Bayesian framework).

Overall, balancing between the size and the robustness of error bounds is an intriguing research challenge. In response to this challenge, the Bayesian-Frequentist-Bound (BFB) framework of [179] offers the possibility to modulate the commitment to prior information, so as to modulate the size of the class of distributions $\mathbb{P}_{g,noise}$ for which the computed bounds are valid at least in a Bayesian-frequentist sense, see the sidebar "A Bayesian-frequentist interpretation of the bounds" for a definition of this concept. In this framework, hyperparameters tuning can also be accommodated to some extent.

KERNEL METHODS AND GAUSSIAN PROCESSES FOR CONTROL

In this section we provide a bird's eye view of the most recent work in learning-based control incorporating the use of kernel-based methods/Gaussian Processes (GPs) within traditional control techniques.

The section is articulated in six paragraphs. In the first paragraph, a Bayesian kernel-based approach (see Section "Regularized least squares") is exploited to identify the system response of a discrete-time and Bounded-Input-Bounded-Output (BIBO) stable linear system. The model obtained is endowed with a detailed description of the uncertainty around it (see Section "Bounds for linear systems") that allows to develop stochastic robust control strategies.

In the subsequent four paragraphs, Gaussian Regression is used to derive state-space models (see box "Gaussian processes and regression: main concepts and formulas" ans sections "Kernel-based estimation of state-space models" and "Bounds for nonlinear system identification"). Within this context, GPs are exploited to provide probabilistic models predicting either the overall dynamics, or a residual uncertainty quantifying the model mismatch with respect to a known nominal description of the system. The confidence of these probabilistic models is an important information that can be used in several ways to properly modify the design of classic model-based control schemes. Specifically, in these four paragraphs,we review the combination of GPs with robust control for linear systems, adaptive control, feedback linearization and model predictive control.

Finally, in the last paragraph we shortly mention the adoption of GPs in model-based reinforcement learning algorithms for control purposes which will be the topic extensively treated in the next section.

Kernel-based stochastic robust control of a SISO linear system

We describe with some details the stochastic robust controller proposed in [56]. The problem regards a discrete-time stable and linear SISO system. Its unknown impulse response is denoted by g with the related transfer function (the z-transform of g) given by $G(z)$. The plant is fed with a known input and has to be estimated from the output data collected in the vector Y. These same data have also to be used to design a controller $C(z)$ such that the closed-loop

Gaussian processes and regression: main concepts and formulas

We have seen the equivalence between kernel-based identification and Gaussian regression (kernels become covariances), see Sections "Kernels and Gaussian regression: an introduction to some key concepts" and "Bayesian interpretation: Gaussian regression of random fields". It is useful to summarise the main formulas starting directly in a stochastic framework since this is the approach taken in several papers combining control and state space models, see Section "Kernel methods and Gaussian processes for control" .

Assume $f : \mathbb{R}^m \to \mathbb{R}$ is a Gaussian process of zero mean and covariance $\lambda\mathcal{K}$, where λ is a positive scale factor and $\mathcal{K}$ is a positive definite Kernel function. This means that, given any finite collection of input locations $\{x_i\}_{i=1}^n$, the sampled function $[f(x_1), \dots, f(x_n)]$ forms a Gaussian vector of covariance $\lambda\mathbf{K}$, where $\mathbf{K}$ is the kernel matrix with (i, j)-entry $\mathbf{K}_{ij} = \mathcal{K}(x_i, x_j)$. Using a standard notation we denote the Gaussian process as

$$f \sim \mathcal{N}(0, \lambda\mathcal{K}).$$

Assume we have a set of n measurements, over the n inputs $\{x_i\}_{i=1}^n$, of the form

$$y_i = f(x_i) + e_i,$$

where $e_i \sim \mathcal{N}(0, \sigma^2)$ is zero mean Gaussian noise with variance σ^2.

As seen, the posterior distribution $p(f|\mathcal{Y})$, is still Gaussian and, using Bayes' rule, one can compute its mean and covariance which give all the information needed to implement an estimator, a predictor and also a control rule. Specifically, the best prediction in the mean squared sense at x^*, given the information $Y = \{y_1, \dots, y_n\}$, is

$$\hat{f}(x^*) = \mathcal{E}\left[f(x^*)|Y\right] = \sum_{h=1}^{n} c_h \, \lambda\mathcal{K}(x_h, x^*),$$

where the coefficients c_h's are given by

$$\begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = (\lambda\mathbf{K} + \sigma^2 I)^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \qquad \mathbf{K} = \begin{bmatrix} \mathcal{K}(x_1,x_1) & \dots & \mathcal{K}(x_1,x_n) \\ \vdots & & \vdots \\ \mathcal{K}(x_n,x_1) & \dots & \mathcal{K}(x_n,x_n) \end{bmatrix}.$$

Moreover, the a posteriori variance of the estimate $f(x^*)$ is given by

$$\mathrm{Var}\left[f(x^*)|Y\right] = \lambda\mathcal{K}(x^*, x^*) -$$

$$\lambda \begin{bmatrix} \mathcal{K}(x_1, x^*) & \dots & K(x_n, x^*) \end{bmatrix} (\lambda\mathbf{K} + \sigma^2 I_n)^{-1} \lambda \begin{bmatrix} \mathcal{K}(x_1, x^*) \\ \vdots \\ K(x^n, x^*) \end{bmatrix}.$$

The above approach can be easily extended to the case where the mean of the Gaussian Process is not zero but equal to a mean function $m(x)$, that is, $f \sim \mathcal{N}(m(x), \lambda\mathcal{K})$. This is useful, since any a priori insight into the dynamics of the system can be readily encoded in the mean function. Indeed it is often possible to capture the main properties of the dynamics, e.g. by using a simple parametric model or a model based on first principles.

Now, consider the dynamical system

$$x_{i+1} = \mathbf{f}(x_i) + \mathbf{e_i} \quad \mathbf{i = 1, \dots, n} \tag{S71}$$

where x_i is the d-dimensional state at instant i, $\mathbf{e_i}$ contains the d random noises and the function $\mathbf{f}$ is vector-valued and encapsulates d transition functions denoted by f_k for $k = 1, \dots, d$.

If the system states are all observable, each transition function f_k can be estimated by modeling it as a Gaussian process and by exploiting the above formulas on the set of pairs $\{x_i, y_{ik}\}_{i=1}^n$, where the measurement y_{ik} is the k-th component of x_{i+1}. It turns out that the overall state transition function $\mathbf{f}$ is estimated employing d independent Gaussian processes. The extension to systems of the form

$$x_{i+1} = \mathbf{f}(x_i, u_i) + \mathbf{e_i} \quad \mathbf{i = 1, \dots, n},$$

where u_i is an input applied to the systems, is easily obtained rewriting the system as $x_{i+1} = \mathbf{f}(\tilde{x}_i)$, $i = 1, \dots, n$, where the augmented states $\tilde{x}_i = (x_i, u_i)$, $i = 1, \dots, n$, are now the input locations to be considered. If a nominal model $\mathbf{f}_{\text{nom}}$ of the system is known a-priori, this knowledge can be incorporated into the mean of the d Gaussian processes. Alternatively, rewriting the system as

$$x_{i+1} = \mathbf{f}_{\text{nom}}(x_i, u_i) + \tilde{\mathbf{f}}(x_i, u_i) + \mathbf{e_i} \quad \mathbf{i = 1, \dots, n},$$

where $\tilde{\mathbf{f}}$ represents the model mismatch between the true model and the nominal model, the previous Gaussian process framework can be applied to directly estimate $\tilde{\mathbf{f}}$.

In recent years, Gaussian regression has been widely adopted to derive state-space models for control purposes. In light of this, in Section "Kernel methods and Gaussian processes for control" we review the use of GPs with traditional control methods, like Model Predictive Control, Robust Control, Adaptive Control, Feedback linearization and Reinforcement Learning.

system

$$\frac{G(z)C(z)}{1+G(z)C(z)}$$

be close to a target transfer function $W(z)$. The distance can be measured by the two-norm

$$\|W(z) - \frac{G(z)C(z)}{1+G(z)C(z)}\|_2. \tag{72}$$

This defines also the fit performance of the controller $C(z)$ applied to the true plant $G(z)$ as

$$100\left(1 - \left\|W(z) - \frac{C(z)G(z)}{1+C(z)G(z)}\right\|_2^2 / \|W\|_2^2\right). \tag{73}$$

Adopting a high-order FIR model with unknown coefficients in the vector θ, system identification can be performed via ReLS (32a). We have also seen that, using its Bayesian interpretation, the obtained model $\hat{\theta}$ can be complemented with a description of the uncertainty given by a Gaussian posterior distribution. In particular, after seeing the data, the impulse response becomes the following Gaussian random vector

$$\theta|Y \sim \mathcal{N}(\hat{\theta}, \hat{\Sigma}) \tag{74}$$

with posterior covariance $\hat{\Sigma}$ given by (64). In (72), we can now replace G with the z-transform of $\theta|Y$. This makes the distance (72) a random variable since it incorporates the stochastic uncertainty around the nominal plant. For any choice of the controller $C(z)$, its probability density can be (in principle) computed using (74). This point is important since it permits to design the controller via statistical criteria. A significant example is given by minimization of the expected value of (72). In general, this however leads to a difficult and nonconvex optimization problem. An interesting convexification is obtained as follows:

- the controller $C(z)$ is linearly parametrized assuming that it is the sum of p basis functions $\phi_i(z)$ with unknown coefficients contained in the vector η, i.e.

$$C_\eta(z) = \sum_{i=1}^{p} \eta_i \phi_i(z); \tag{75}$$

- the performance index (72) is so reformulated: for any plant g it is given by

$$\mathrm{Err}_\eta(g) := \left\|W(z)(1+G(z)C_\eta(z)) - G(z)C_\eta(z)\right\|_2^2. \tag{76}$$

Now, we replace g with the Gaussian vector $\theta|Y$ and take the expectation so that the objective depends only on the controller parameters η. Hence, our *stochastic robust control* problem becomes

$$\hat{\eta} = \arg\min_{\eta} \ \mathcal{E}[\mathrm{Err}_\eta(\theta|Y)]. \tag{77}$$

Interestingly, as shown in [56], the optimization problem (77) is quadratic in η and thus admits a closed form solution. In fact, it is equivalent to solving

$$\arg\min_{\eta} \ \mathrm{Err}_\eta(\hat{\theta}) + \eta^\top A \eta \tag{78}$$

where, recalling (74), $\hat{\theta}$ is the estimate of the plant (the posterior mean of θ) while A is a suitable matrix which depends on the posterior covariance $\hat{\Sigma}$ and the basis functions ϕ_i that generate the controllers' space, see [56] for details. One can thus see that the optimal coefficients in η trade-off two different terms: the error relative to the nominal plant plus another one that accounts for its uncertainty (with A that can be interpreted as a regularization matrix).

Instead of minimizing the expectation of the distance (76) as in (77), other forms of robustness could be pursued. Another example in [56] regards minimization of the worst-case distance using a minmax formulation coupled with the scenario approach [166].

A numerical experiment

For illustrative purposes, we consider a benchmark example taken from [56]. A realistic posterior distribution (74) of linear dynamic systems is built as follows. The mean $\hat{\theta}$ is given by the first 200 impulse response coefficients of the following rational transfer function

$$\bar{G}(z) = \frac{0.28261z + 0.50666}{z^4 - 1.41833z^3 + 1.58939z^2 - 1.31608z + 0.88642} \tag{79}$$

while the covariance $\hat{\Sigma}$ is obtained by an identification experiment. In particular, 500 output measurements are obtained applying to $\bar{G}$ an input given by white noise filtered by a randomly generated 2nd order stable transfer function. The outputs are then corrupted by white Gaussian with a signal-to-noise-ratio (SNR) equal to 100. A FIR model of dimension 200 is obtained by ReLS (32a) using the stable spline prior (36) with hyperparameters estimated via marginal likelihood optimization. The posterior covariance $\hat{\Sigma}$ is then computed using (64).

Now, we consider a Monte Carlo study where at any run a plant $G(z)$ (represented by a FIR of dimension 200) is drawn from our Gaussian posterior distribution of mean $\hat{\theta}$ and covariance $\hat{\Sigma}$. The controller $C(z)$ is a FIR of order 5 combined with an integrator:

$$C_\eta(z) = \frac{\eta_1 + \eta_2 z + \ldots + \eta_6 z^5}{z^4(z-1)} \tag{80}$$

with η obtained using two different algorithms. The first one, called *Nominal*, determines the parameter vector η as the minimizer of $\mathrm{Err}_\eta(\hat{\theta})$ with the objective defined in (76). So, no uncertainty is included in the controller synthesis, just the posterior mean is used. The second one, called *Robust*, achieves η by minimizing (78). Hence, it exploits the uncertainty around $\hat{\theta}$ and minimizes the expected error. Fig. 11 (left panel) shows the fits (73) achieved by the two algorithms after 300 runs. The control performance is largely improved by exploiting the Gaussian uncertainty bounds around the nominal model. The right panel of the same figure shows the results obtained by reducing uncer-

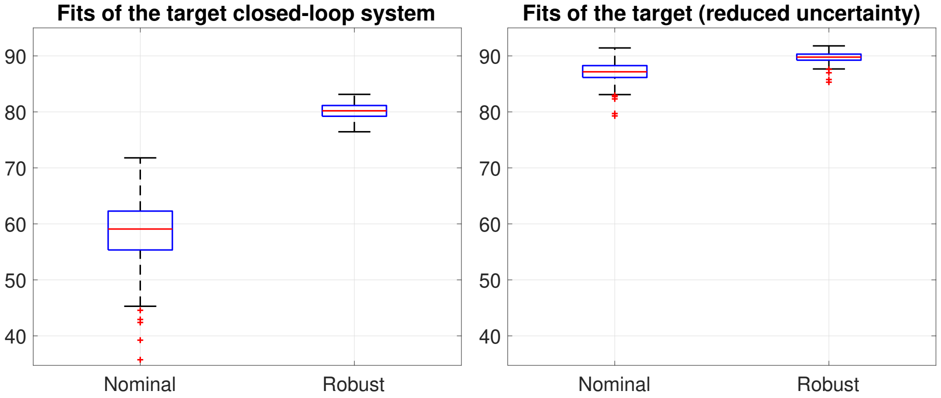

Figure 11 *Robust control of a SISO linear system.* Fits of the closed-loop system. Different open-loop systems are drawn from a Gaussian random vector with posterior covariance defined by the stable spline prior (36) conditional on the available output data. The controller is then designed using only the mean of the random vector (*Nominal*) or taking into account the uncertainty by minimizing the average distance (77) (*Robust*). Results in the right panel are obtained by reducing the uncertainty around the posterior mean (the posterior covariance is divided by a factor 10).

tainty, repeating the Monte Carlo study with $\hat{\Sigma}$ divided by 10.

GPs - based Model Predictive Control

The combination of MPC and learning techniques is a research field that is attracting a great interest in the control community, because of the opportunity of a data-driven improvement of the closed-loop action, while maintaining the established guarantees of optimal control [64].

The use of GPs as prediction models in the MPC framework is dated back to 2005, when in [223] the authors have described a general Nonlinear Model Predictive Control (NMPC) algorithm based on a Gaussian Process model. In [223], the proposed approach has been tested on a benchmark pH process control. However the trend of most recent research (see e.g. [62], [63]) is that of exploiting GPs as an augmentation for a physics-based model in order to estimate unexpected disturbances rather than learning the overall plant dynamics, see for example [224].

In general, the optimization problem arising in GPs-based MPC can be formulated as a Stochastic Optimal Control Problem (SOCP), where the minimization function is an expected value, the differential equation constraint (i.e. the dynamics of the system) is subject to uncertainties and the system constraints must be satisfied in probability. The direct solution of SOCP is computationally hard, especially when dealing with non-linear systems. In such case, the main challenge is the uncertainty propagation over the prediction horizon since Gaussian uncertainty (obtained from the GP) is no longer Gaussian when propagated through the nominal nonlinear dynamics. As a consequence approximation methods are needed, such as exact moment matching [225], linearization [62], ellipsoidal uncertainty set propagation [226] or sigma-point transform [227]. In addition, further approximations are adopted (e.g., only the mean of the process is propagated or the GP's covariances are kept constant over the prediction horizon [62]) to reduce the computational costs and to achieve real-time implementation. A side effect of all these approximations is that crucial information of the model is lost and the probabilistic constraints might be violated.

Recently in [228], an open-source toolbox combining a MATLAB-based Fast NMPC solver and a Python library for Gaussian Process regression has been presented to define an off-the-shelves framework for implementation of GP-based Learning-based NMPC. Starting from a nominal model and model mismatch data or input-output measurements, the toolbox allows to train the GP either as model mismatch estimator or black-box model and get automatically the information needed for the NMPC problem.

Feedback linearization using Gaussian Processes

Feedback linearization is a general technique employed to control nonlinear systems. It consists in transforming a nonlinear control system into an equivalent linear control system through a change of variables and a suitable control input. Consider a system of the form

$$\dot{x} = f(x) + G(x)u, \tag{81}$$

where $x, u \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}$ and $G \in \mathbb{R}^{n \times n}$. For simplicity, assume that G is invertible for any x. Then, by applying $u = G^{-1}(x)\left(-f(x) + a\right)$ where a is an auxiliary input, the resulting dynamics turns out to be $\dot{x} = a$. Now, depending on the specific task to be accomplished, a can be efficiently designed (in particular guaranteeing exponentially fast convergence) resorting to classical feedback control techniques used for linear systems. Typically, we refer to the design of u and a as, respectively, the inner loop and the outer loop of the overall control scheme. Notice that, the feedback linearization approach above reviewed, requires the accurate knowledge of the model.

In [60] the authors have considered a system of the type (81) written in the controllable canonical form (see Eq. (1) in [60]) with the goal of stabilizing it. In [60], both f and G are assumed to be unknown, and estimates $\hat{f}$ and $\hat{G}$ are obtained using Gaussian Processes. In the identification procedure, the knowledge of the control affine structure of the system is transferred into the kernel function, allowing to identify system in closed-loop, while an arbitrary controller is running the system. Based on $\hat{f}$ and $\hat{G}$ the following controller is considered

$$u = \hat{G}^{-1}(x)\left(-\hat{f}(x) + a\right) \tag{82}$$

where the auxiliary input a is designed in such a way to drive x to 0 exponentially fast if $\hat{f}$ and $\hat{G}$ coincide with f and G, respectively. As main contribution, it is proved that (82) is globally uniformly bounded and that the ultimate bound is reduced as more knowledge (training data) is available. Interestingly an upper bound on the size of the set to which the system converges with a probability greater than a given threshold is derived depending on the maximum mutual information that can be extracted from a training set composed by an assigned number of points, and on the covariance of the estimate obtained on the function f. In particular the smaller the mutual information and the covariance, the smaller the upper bound. A similar approach has been proposed also in [229] in the context of control of mechanical systems.

A robust version of the learning-based feedback linearization strategy previously described has been proposed in [61] in the context of tracking control of Lagrangian systems. In [61], the authors have assumed to have a nominal knowledge of the system and GPs are used to approximate the error between the commanded acceleration and the actual acceleration of the system. The predicted mean and variance of the GP are used to calculate an upper bound on the uncertainty of the linearization, which, in turn, is used in the design of a term to be added to the outer controller a to make robust the overall feedback linearization scheme. It is proved that the proposed strategy, guarantees that the tracking error converges to a ball with a radius that can be made arbitrarily small through appropriate control design.

Integrating Gaussian Processes and Adaptive Control

Traditionally, adaptive control deals with systems with parametric uncertainties. Of notable mention are dynamic models that are affine in the input and in the uncertain parameters, that is,

$$\dot{x} = f_x(x) + f_u(x)u + f_\theta(x)\theta$$

where x is the state, f_x, f_u and f_θ are nominal functions assumed to be known, u is the input and θ is the vector parameters uncertainties. The control input is $u(t) = \pi(x(t), \hat{\theta}(t))$, where $\hat{\theta}(t)$ is an estimate of θ. Adaptive control aims at adjusting online the vector $\hat{\theta}$ (and, in turn, the characteristics of the controller) based on the output feedback of the system in a way that the tracking error is reduced while stability is maintained. The vector $\hat{\theta}$ is typically adapted by using either a Lyapunov function to guarantee that the closed-loop system is stable or Model Reference Adaptive Control (MRAC) to make the uncertain controlled system tracking the behavior of a predefined stable reference model [230].

One of the main challenges in adaptive control is preventing the estimated model to overfit to the latest observations, [230], [231]. GP-based probabilistic models provide a useful tool in this regard and are exploited by learning-based adaptive control approaches to achieve cautious adaptation by weighting the contribution of the learned model based on the model output uncertainty.

An interesting approach has been proposed in [58] where the authors have considered the system $\dot{x} = f(x,u) + \tilde{f}(x,u)$ where $f(x,u) = f_x(x) + f_u(x)u$ is the known nominal dynamics and where $\tilde{f}(x,u)$ is the unknown dynamics of the same form, that is, $\tilde{f}(x,u) = \tilde{f}_x(x) + \tilde{f}_u(x)u$, being $\tilde{f}_x(x)$ and $\tilde{f}_u(x)$ unknown nonparametric nonlinear functions. In the context of MRAC, the adaptive law introduced in [58] is given by the sum of two suitably weighted terms. The first term, denoted as $\pi_{\text{nom}}(x(t))$, is a control policy derived from the application of a feedback linearization approach to the nominal model. Due to the unknown dynamics, $\pi_{\text{nom}}(x(t))$ introduces feedback linearization errors that are compensated by the second term, denoted as $\pi_{\text{learn,t}}(x(t))$, which is the adaptive component designed based on GP approach. To deal with the uncertainty of the GP model learning, the two terms are combined as $\pi_t(x(t)) = \pi_{\text{nom}}(x(t)) - \gamma(x(t), u(t))\,\pi_{\text{learn,t}}(x(t))$ where $\gamma(x(t), u(t)) \in [0,1]$ is a weighting factor, with $\gamma = 0$ denoting low confidence in the GP. A stochastic stability analysis has proved the stability of the overall system. The effectiveness of the approach has been tested in quadrotor experiments [58], [59].

Interestingly, in [232], the authors have proposed an approach to the training of GP models for MRAC inspired by generative neural networks models. The architecture introduced in [232] is termed as Model Reference Generative Network (MRGeN). Loosely speaking MRGeN is a neural network model for the system uncertainties, which predicts the pair of state-uncertainties for GP inference. The MRGeN weights are updated such that network weights are moved in the direction of reducing the reference model tracking error.

Gaussian Processes models for Robust linear control

Robust control is another control design technique that deals with uncertainty. Robust control methods are designed to function properly provided that uncertain parameters or disturbances are found within some, typically compact, set [233]. Differently adaptive control, which adapts to the parameters currently present, the goal of robust control is to find a suitable controller for all possible disturbances and to keep the controller unchanged after the initial design. Consider time-invariant systems composed by the sum of a known linear nominal model and an unknown nonlinear component, that is,

$$x_{k+1} = Ax_k + Bu_k + \tilde{f}(x_k, u_k, w_k) \tag{83}$$

where w_k is a process noise and where $\tilde{f}(x_k, u_k, w_k) \in \mathbb{F}$, with $\mathbb{F}$ being known and bounded. Quite often in literature it is assumed that $\tilde{f}(x_k, u_k, w_k) = \tilde{A}x_k + \tilde{B}u_k + w_k$ where $\tilde{A}$, $\tilde{B}$ are unknown matrices. Specific structures of $\tilde{A}$, $\tilde{B}$ allow to model different types of uncertainties, e.g., additive, multiplicative and feedback uncertainties. Design techniques, such as H_∞ and H_2-control design, yield controllers that are robustly stable for all $\tilde{f} \in \mathbb{F}$.

It is known that classical robust control approaches might attain conservative performance in particular when the uncertainty region is quite large. The goal of learning-based robust control is to improve the performance by reducing the model uncertainty in Eq. (83).

In [55], the unknown nonlinear dynamics $\hat{f}$ are learnt as a GP, which is then linearized about an operating point. Specifically, the uncertain linear dynamics in (83) are assumed to be modelled as $(\tilde{A}_0 + \tilde{A}_1 \circ \Delta_A)x_k + (\tilde{B}_0 + \tilde{B}_1 \circ \Delta_B)u_k$, where $\tilde{A}_0$ and $\tilde{B}_0$ are obtained from the linearized GP mean, $\tilde{A}_1$ and $\tilde{B}_1$ are obtained from the linearized GP variance (often two standard deviations), and Δ_A and Δ_B represent matrices with elements taking any value in the range of $[-1, +1]$. The discrete-time controller is designed by solving a suitable convex optimization problem in terms of linear matrix inequalities where the objective is to minimize an error signal caused by all the possible uncertainties. The proposed approach has been tested on a quadrotor.

Reinforcement Learning

Another field where GPs are used in control applications is Reinforcement Learning (RL). RL is based on the the idea of learning a control law to achieve a task by interaction with the surrounding world. RL algorithms can be categorized in multiple sub classes accordingly to which characteristics of the algorithms we want to focus on. For example one of the most common distinctions is between model-free and model-based algorithms. In the first class of algorithms, GPs are commonly used to approximate the value function: [234] introduced to use GP to learn the temporal difference og the value function in GP-TD, then refined in GP-SARSA [235], and then made more data efficient using delayed GP updates in DGPQ [236] and by combining demonstrations and exploration techniques in GPPSTD [237]. In GPQ-MFRL [238] the authors propose to use incrementally more difficult simulators to learn the value function with GPs. Instead, in model-based algorithms, GPs are used to build a model of the system dynamics based on data collected interacting with the system. This class of algorithms has been successfully applied to solve control applications on mechanical system, and it will be discussed in the next section. There are also hybrid approaches that for example try to combine both model-based and model-free algorithms like in [239] or where model-based reinforcement algorithms are combined with standard control techniques like in [50].

MODEL-BASED REINFORCEMENT LEARNING

Model Based Reinforcement Learning (MBRL) algorithms collect data by interacting with the environment to learn a dynamical model of the system. As in MPC, such model is used to simulate the system and update the policy by optimizing the simulated dynamics. In this way, MBRL algorithms aim at limiting interaction time with the real system and improving data efficiency. However, differently from MPC, which optimizes the future evolution of the system online, MBRL algorithms perform simulation and control optimization offline, getting over the computational time constraints due to real-time control. This difference makes the usage of more complex model possible and allows considering stochastic behaviors in a comprehensive way.

MBRL algorithms can be divided into value-based and policy-search-based algorithms. The first example of MBRL algorithm can most likely be attributed to [240] under the name of Dyna. The Dyna architecture proposes a general scheme for value-based algorithms which use the accumulated experience to simultaneously build a dynamical model of the system, update the value function, and as consequence, learn a policy. The Dyna algorithm proposes to update the value function applying a Q-learning approach (or any other suitable approach) in the learned dynamical model instead of the actual system. Some relevant evolution of this approach are proposed in VAML [241] and IterVAML [242] where notably the modeling and planning part of the algorithm are not independent and the value function is considered while learning the model.

The second category of MBRL, namely policy-search-based algorithms, is the one we will focus on the most because it has a successful history of real world applications to mechanical systems and it is explored both in the RL and in the control community. A pioneer algorithm was proposed in [65], [243], [244] under the name of Probabilistic Inference for Learning Control (PILCO), which inspired several MBRL algorithms e.g., [67], [245], [246], [68], [69],

[247].

Given the system state $x_t \in \mathbb{R}^d$ and the system input $u_t \in \mathbb{R}^m$, these algorithms model the system dynamics as a discrete-time system with an unknown one-step-ahead stochastic transition function $f(\cdot)$. Let $\tilde{x}_t = [x_t, u_t]$ be the augmented state concatenating x_t and u_t; then, we have

$$x_{t+1} = f(\tilde{x}_t).$$

The applied inputs are selected according to a policy function $\pi_\theta(x_t)$ that depends on the state x_t and the policy parameters θ. For instance, a widely used policy in MBRL is the RBF-network policy, followed by a squashing function to limit system inputs if necessary. The parameters of the RBF-network are centers, lengthscales and weights of the Gaussian functions, denoted, respectively, by a, l, and w, i.e., $\theta = \{a, l, w\}$. The expression of a RBF-network policy with n_g basis is

$$\pi_\theta(x_t) = \sum_{i=1}^{n_g} w_i \exp\left(-\sum_{j=1}^{d}\left(\frac{a_i^j - x_t^j}{2l^j}\right)^2\right). \tag{84}$$

Other examples of policy functions can be the linear policy, the PID controller, or general ANN.

In this class of algorithms a cost function $c(x_t)$ encodes the task to be accomplished. For instance, a widely used cost function adopted in MBRL is the saturated distances from the target state x_*, expressed by the following equation,

$$c(x_t) = 1 - e^{-(x_t - x_*)^T L (x_t - x_*)}, \tag{85}$$

where L is a diagonal matrix. The diagonal elements of L allow weighting distances w.r.t. the different state components. $c(x_t)$ defines the instantaneous cost, the actual objective function optimized by this class of algorithms is the expectation of the cumulative cost, i.e., the sum of the costs occurred in T steps, expressed as

$$J(\theta) = \sum_{t=0}^{T} \mathcal{E}(c(x_t)). \tag{86}$$

The expectations in (86) are computed w.r.t. the state distribution induced by the initial distribution $p(x_0)$, $f(\cdot)$, and θ.

The general algorithmic structure followed by this class of algorithms consists of a repetition of several attempts to solve a desired task, called trials. For each trial the following three steps are computed:

» *Model Learning*: the data collected from all the previous interactions are used to build/update $f(\tilde{x}_t)$, the stochastic model of the one-step-ahead transition function (at the first trial, data are collected applying possibly random exploratory controls);

» *Policy Update*: the policy parameters θ are optimized in order to minimize $\hat{J}(\theta)$, that is an approximation of $J(\theta)$.

» *Policy Execution*: the current optimized policy is applied to the system and the data are stored for model improvement.

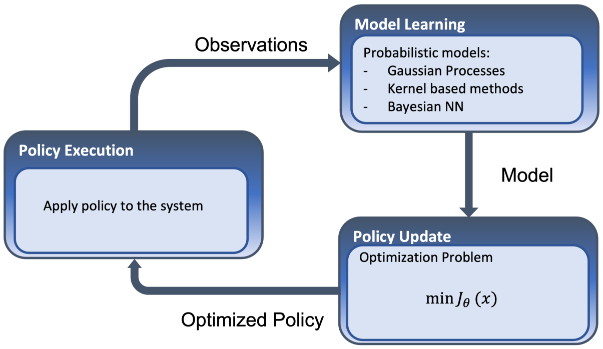

Figure 12 Illustration of the MBRL main three computational steps at each trial.

Figure 12 represents one trial of a typical MBRL algorithm where the 3 steps described above are repeated in cycle to improve the model and the policy of the algorithm.

In the next sections, we enter more in details of some of the main policy-search-based algorithms dividing them accordingly to how the state distribution is propagated, either with moment matching or with particles evolution.

PILCO and Approaches based on Moment Matching

As mentioned above, the algorithm PILCO [65] is recognized as one of the most fundamental and representative algorithm of the policy-search class of MBRL algorithms. For this reason, we describe the realization of the above general three steps proposed by PILCO as baseline. This description will be then helpful in the discussion on the main MBRL algorithms developed in the wake of PILCO.

PILCO relies on GPR to learn the transition function $f(\tilde{x}_t)$. Each of the d components of $f(\tilde{x}_t)$, hereafter denoted by $f^i(\tilde{x}_t)$, with $i = 1 \dots d$, is modeled with a distinct GP, independent from the others given the GP input $\tilde{x}_t$. The algorithm assumes that the state is completely observable, i.e., we measure all the components of x_t, and derives the input-ouput dataset used to train GPs starting from the state-input dataset $\mathcal{D} = \{(x_t, u_t), t = 1, \dots, Tn\}$ collected in previous n trials. The prior of each i-th GP is normally distributed with mean x_t^i, i.e., the i-th state component at the current time t for better numerical properties and covariance matrix defined by a kernel function $k^i(\tilde{x}_j, \tilde{x}_h)$ given by the sum of two terms, namely,

$$k^i(\tilde{x}_p, \tilde{x}_q) = k_G^i(\tilde{x}_p, \tilde{x}_q) + \delta_{pq}\sigma^2, \tag{87}$$

where k_G^i is the Gaussian kernel defined in (58), while $\delta_{pq}\sigma^2$, with $\delta_{pq} = 1$ (resp. 0) if $p = q$ (resp. $p \neq q$) is the regularization term needed to account for noise. Then the posterior distribution of x_{t+1} given the dataset $\mathcal{D}$ and a general augmented GP input $\tilde{x}_t$ is Gaussian distributed, namely,

$$p(x_{t+1}|\tilde{x}_t, \mathcal{D}) \sim N(m_f(\tilde{x}_t), \Sigma_f(\tilde{x}_t)). \tag{88}$$

In the previous equation $m_f(\tilde{x}_t)$ and $\Sigma_f(\tilde{x}_t)$ denote, respectively, the posterior mean and variance, and they are given by the following expression,

$$m_f(\tilde{x}_t) = [m_f^1(\tilde{x}_t) \ldots m_f^d(\tilde{x}_t)]^T,$$
$$\Sigma_f(\tilde{x}_t) = \mathrm{Diag}(\sigma_f^1(\tilde{x}_t) \ldots \sigma_f^d(\tilde{x}_t)),$$

where each $m_f^i(\tilde{x}_t)$ and $\sigma_f^i(\tilde{x}_t)$ are computed according to the formulas reported in the box "Gaussian processes and regression: main concepts and formulas".

At each optimization step of the *Policy Update* the algorithm has to compute $J(\theta)$. The expectations in (86) require the state distributions $p(x_0), \ldots p(x_T)$ induced by θ and the one-step-ahead model $f(\cdot)$. Specifically, starting from the initial distribution $p(x_0)$, for each time step t, the distribution of x_{t+1} is obtained by marginalization of (88), namely,

$$p(x_{t+1}) = \int p(x_{t+1}|\tilde{x}_t, \mathcal{D}) p(x_t, \pi_\theta(x_t)) dx_t. \tag{89}$$

Unfortunately, the exact computation of the previous integral is not tractable. PILCO assumes that all the distributions are Gaussians and obtains an analytical approximation of the integral in (89) relying on moment matching. First, moment matching is applied to approximate the state-control joint distribution $p(x_t, \pi_\theta(x_t))$. Then, the same procedure is applied to (89). Finally, the algorithm approximates $J(\theta)$ computing the expectations in (86) using the Gaussian approximation of $p(x_0), \ldots p(x_T)$. As showed in [243], if the cost function has a structure of the kind reported in (85) the integrals in (86) is analytically tractable, and the approximation of $J(\theta)$ together with its gradient w.r.t. θ are derived in closed form. Then, the control parameters θ are optimized with a gradient-based optimization.

The effectiveness of PILCO has been demonstrated in several experiments, both in simulated and real setups, ranging from low dimensional tasks, such as the pendulum and cart-pole swing-up, and the throttle valve control, to higher dimensional tasks, such as the unicycle stabilization and robotics manipulation. However, the analytical approximation of $J(\theta)$ introduces several limitations:

» The computation of the moments required to apply moment matching is tractable only when considering the Gaussian kernel as prior in (87), and cost functions for which the integrals in (86) is analytically tractable.

» Moment matching forces all the distributions to be Gaussian, consequently, the state distributions are unimodal, which might be too crude an assumption on the long-term system dynamics for several systems.

The limitation on the kernel choice might be very stringent in certain applications, as the Gaussian kernel assumes that the underlying process is stationary and smooth. Such properties are not always met in the actual system, for instance in mechanical systems. This mismatch might lead to overfitting, besides limiting generalization properties in data that have not been seen during training [248], [249], [45], [46].

A solution to the poor generalization properties of the Gaussian kernel in unexplored data has been proposed in [66], where the GP model is trained with data coming from a simulator before starting the actual reinforcement learning procedure. The experience on the simulator improves the performance of PILCO in areas of the state space with no available data points. However, the effectiveness of this method depends on the accuracy of the simulator, which may not always be available.

Another alternative approach developed in the wake of PILCO is Deep-PILCO [67]. To overcome assumptions on stationary and smoothness imposed by the Gaussian kernel, the algorithms relies on Bayesian Neural Networks [250] to learn the system dynamics, so that the model can capture also non-stationary and discontinuous dynamics. Long term distributions are approximated with moment matching, with the moments computed via Monte Carlo simulation. Experiments shows that, compared to PILCO, Deep-PILCO requires a larger number of interactions with the system in order to learn the tasks, due to the high dimensional model parametrization. Similar results have been highlighted also by experiments carried out in [245], here the authors introduced a more recent NN-based MBRL algorithm named PETS. In low-dimensional tasks GP-based algorithms outperform the ones based on NN. Instead, when the state dimension and the number of samples grow, the straightforward application of GP-based approaches might be critical and less effective than NN-based approaches.

Particles-based approaches

The algorithms mentioned above in Section approximate the long-term state distributions relying on moment matching. As mentioned before, the moment matching approximation proposed in PILCO imposes the use of the Gaussian kernel and unimodal state distributions. The unimodal approximation could be a too crude assumption on the long-term system dynamics. Moreover, it introduces relevant limitations in case that initial conditions or the optimal solution are multimodal. For instance, in case that the initial variance of the state distribution is high, the optimal solution might be multimodal, due to dependencies on initial conditions.

An alternative route to moment matching to approximate the long-term state-input distributions relies on particles based approach. Given the policy parameters θ and the transition model $f(\cdot)$, the integral in (86) is approximated simulating the evolution of a batch of m particles. The process starts by sampling the batch of particles states $\mathcal{Z}_0 = \{z_0^1 \dots z_0^m\}$ from the initial state distribution $p(x_0)$. Then, at each simulation step t, the algorithm evaluates the policy to compute the control input of each particle and samples $\mathcal{Z}_{t+1} = \{z_{t+1}^1 \dots z_{t+1}^m\}$ from (88). Once that the $\mathcal{Z}_0 \dots \mathcal{Z}_T$ are given the expectations in (86) are approximated by the algebraic mean, namely,

$$\hat{J}(\theta) = \sum_{t=0}^{T} \frac{\sum_{i=1}^{m} c(z_t^i)}{m}. \tag{90}$$

A first attempt based on this approximation has been proposed in [246]. The authors relied to a gradient-based optimization strategy to approximate θ. The gradient is computed using the strategy proposed in PEGASUS [251], where by fixing the initial random seed, a probabilistic Markov decision process (MDP) is transformed into an equivalent partially observable MDP with deterministic transitions. The authors highlighted several limitations due to the inability of the gradient-based optimization to escape from numerous local minima generated by the multimodal distribution. Compared to PILCO, results obtained were not satisfactory.

An alternative solution to compute the gradient from particle-based approximation is the *reparameterization trick*, successfully introduced in stochastic variational inference (SVI) [252], [253]. The computation of (90) involves stochastic operations, consequently $\nabla_\theta \hat{J}(\theta)$, the gradient of $\hat{J}(\theta)$ w.r.t. θ, can not be computed straightforwardly by back-propagation. The *reparameterization trick* re-defines the stochastic operations so that sampling is differentiable w.r.t. x_t, u_t, and also θ. Given the particle $i = 1 \dots m$ at simulation step t, instead of sampling z_{t+1}^i directly from (88), the *reparameterization trick* first samples a point ϵ from a zero-mean and unit-variance normal distribution. Then, ϵ is mapped into the distribution defined by (88) applying the following standard expression,

$$z_{t+1}^i = m_f(\tilde{x}_{t+1}) + L_{t+1}\epsilon,$$

where L_{t+1} is the Cholesky decomposition of $\Sigma_f(\tilde{x}_{t+1})$, namely, $\Sigma_f(\tilde{x}_{t+1}) = L_{t+1}L_{t+1}^T$. In this way, $\nabla_\theta \hat{J}(\theta)$ can be computed directly by backpropagation applying the chain rule.

The algorithm PIPPS [68] experimented with the *reparameterization trick* to estimate the gradient, highlighting several issues due to exploding magnitude and random direction. To overcome such limitations PIPPS proposed the *total propagation algorithm*, which regularized the gradient obtained with the *reparameterization trick* using the *likelihood ratio* gradient [254]. PIPPS performs similarly to PILCO with some improvements both in the gradient computation and in the overall performance when the level of noise increases.

A recent work based on the *reparameterization trick* is MC-PILCO [69], which follows a different approach to avoid issues due to exploding magnitude and random direction in the gradient computation. The authors show that by introducing the dropout during policy optimization and by shaping the cost function opportunely the *reparameterization trick* can actually be used to compute the gradient in particle-based GP MBRL algorithms and Monte Carlo methods do not suffer of gradient estimation problems. The use of dropout that was introduced in the deep learning community [255] to avoid overfitting during training Deep Neural Networks, is revisited in a control framework in [69] to optimize the control policy. This makes the policy stochastic during learning which increases the entropy of the particles distributions and helps the optimization algorithm to escape local minima in the parameter space.

One of the advantages of particle-based approaches in MBRL is that it is possible to remove all the kernel assumptions that were required to compute closed form expressions of the gradients when using moment matching. Indeed, the advantages of using kernels with more structured than the limited Gaussian kernel are demonstrated both in simulation and on real systems in MC-PILCO. Finally, MC-PILCO is extended to cope with systems with Partially Measurable States, MC-PILCO4PMS. Considering for example a mechanical system, it is likely that only positions are actually measurable in the real system, while other components of the state like velocities and accelerations are only numerically derived with filters from the history of the positions. This fact leads to a differentiation between the states available during policy execution, which need to be computed with fast online filters, and the states available during offline learning where the states can be computed in a non-physical way to improve the accuracy. MC-PILCO4PMS proposes both to learn the GP models using accurate a-causal filter to improve the long-term predictions and to simulate the online observation system during particles propagation in policy optimization. The latter effectively injects additional noise to the model predictions to emulate the state which will be seen during policy execution. Recently, the same authors proposed a variation of the algorithm MC-PILCO4PMS, specifically designed for mechanical systems when the joint velocities are not available in [247].

Finally, Black-DROPS [70] is another particle-based approach, which mainly differs from the above methods because it uses a gradient-free policy optimization to avoid gradient estimation issues. The main advantages are that there are no constraints in the type of cost function considered in algorithm, which can even be non-differentiable, the policy optimization relies on robust black-box algorithms such as CMA-ES [256] in order to escape from local minima and the data efficiency of the algorithm is comparable to analytical approaches such as PILCO. Furthermore, this approach carries the advantage that the optimization can be parallelized in modern GPU clusters. Black-DROPS achieves similar data efficiency to PILCO's, but significantly increases asymptotic performance, thanks to the better accuracy of particle-based approximation, and the ability of the gradient-free optimizer to escape from local minima.

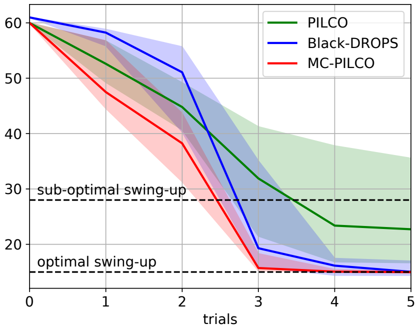

Figure 13 Median and confidence intervals (25-75%) of the cumulative cost as a function of trials obtained with PILCO, Black-DROPS and MC-PILCO. Success rates are reported below.

Success Rates

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
PILCO	2%	4%	25%	34%	38%
Black-DROPS	0%	4%	33%	69%	84%
MC-PILCO	0%	14%	70%	98%	100%

Experiments

In this section, we report a comparison between three of the MBRL algorithms previously discussed, namely, PILCO, Black-Drops, and MC-PILCO. We considered these algorithms not only for their importance, but also because they made the source code available for comparison. Besides that, we present an application of MC-PILCO carried out on a real setup, highlighting the benefits due to the the possibility of including prior knowledge in the kernel function.

We compared the three algorithms on the simulated cart-pole swing-up task, which is a standard benchmark both in the control and RL community. Indeed, despite the system is low-dimensional, this benchmark is particularly hard due to the under-actuation and the highly nonlinear dynamics. The system consists of a cart and a pole. The cart is constrained by a rail to move horizontally, while a non-actuated revolute joint connects the cart and the pole, so that the pole rotation plane is perpendicular to the ground. The state of the system is given by $p[m]$ and $\theta[rad]$, i.e., the cart position and the pole angle, together with their time derivatives. When the pole is in the downward stable equilibrium $\theta = 0$, while the unstable equilibrium point is in $\theta = \pi$. The control action is the force that pushes the cart horizontally. The goal is to swing-up the pendulum and keep it in the unstable equilibrium point starting from the initial state distribution $\mathcal{N}([0,0,0,0], \text{diag}([10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}]))$. The geometrical and dynamical properties of the system are the same as the system used in PILCO [65]. The sampling time and the control period are $T = 0.05$ seconds. The state measurements are corrupted by an i.i.d Gaussian noise with standard deviation 10^{-2}.

We implemented a Monte-Carlo study to compare the three algorithms. For each algorithm we run 100 experiments on the simulated cart-pole task. Every experiment is composed of 5 trials, each of length 3 seconds. The random seed varies at each experiment, corresponding to different exploration data and initialization of the policy, as well as different measurement noise realizations. The policies optimized by the algorithms are RBF-networks like the one in (84). The three algorithms adopted cost functions of the kind reported in (85) to encode the task, with some minor differences to accommodate the different strategies used for approximation and optimization. All the results are reported w.r.t. the PILCO cost function, that is

$$c^{\text{pilco}}(x_t) = 1 - \exp\left(-\frac{1}{2}\left(\frac{d_t}{0.25}\right)^2\right), \qquad (91)$$

where d_t^2 is the squared euclidean distance between the tip of the pole and its position at the unstable equilibrium point with $p_t = 0$ [m]. Finally, as regards model learning, all the algorithms defines the GP prior as in (87).

Figure 13 reports the median of the cumulative costs collected in the Monte Carlo experiment as a function of the trials, namely, the experience accumulated on the system. We also reported the first and third quartiles to provide a measure of the dispersion around the median. The table below Figure 13 reports the success rates collected at each trial, namely the percentage of "success" collected in the 100 experiments. We label a trial as "success" if $|p_t| < 0.1$ [m] and 170 [deg] $< |\theta_t| <$ 190 [deg] $\forall t$ in the last second of the trial. In this task, MC-PILCO achieved the best performance both in transitory and at convergence, as demonstrated by the evolution of its cumulative cost distribution, which is lower than the ones of PILCO and Black-Drops. Similar considerations can be draw by comparing the success rates: at trial 4 and 5 MC-PILCO success rates are, respectively, 98% and 100%, while the ones of PILCO and Black-DROPS are still far from 100%. PILCO showed poor convergence properties, since at trial 5 the success rate is only 38%, and the cumulative cost dispersion around the median is still considerable compared to MC-PILCO and Black-DROPS. Black-DROPS outperforms consistently PILCO at trial 3, 4 and 5, but without reaching MC-PILCO performance.

We applied MC-PILCO in real setup to solve a swing-up task. Instead of using a cart-pole, we considered a Furuta pendulum (FP) [257]. The FP is a popular benchmark in nonlinear control and RL. The system is composed of three links and two revolute joints. The first link, named base, is fixed and perpendicular to the ground. The second link, named arm, rotates on a plane parallel to the ground, while the rotation axis of the last link, the pendulum, is parallel to the principal axis of the second link. A picture of the system is reported in Figure 14. Like the cart-pole, the FP is an under-actuated system since only the first joint is actuated through a DC motor. The control input is the voltage of the DC motor. The angles of the horizontal and vertical joints, hereafter denoted θ^h and θ^v, are measured by optical encoders with 4096 [ppr] at 30 [Hz]. The task consists in learning a controller able to swing-up the pendulum in the unstable equilibrium point ($\theta_h = 0$, $\theta_v = \pm\pi$) starting from the $\theta_h = 0$ and $\theta_v = 0$. The cost function is given by the following expression,

$$c(x_t) = 1 - \exp\left(-\left(\frac{\theta_t^h}{2}\right)^2 - \left(\frac{|\theta_t^v| - \pi}{2}\right)^2\right) + c_b(x_t), \tag{92}$$

with

$$\begin{aligned} c_b(x_t) = & \frac{1}{1 + \exp\left(-10\left(-\frac{3}{4}\pi - \theta_t^h\right)\right)} \\ & + \frac{1}{1 + \exp\left(-10\left(\theta_t^h - \frac{3}{4}\pi\right)\right)}. \end{aligned}$$

The first part of the function in (92) promotes solutions that reach the target state $\theta_t^h = 0$ and $\theta_t^v = \pm\pi$, while $c_b(x_t)$ penalizes trajectories where $\theta_t^h \leq -\frac{3}{4}\pi$ or $\theta_t^h \geq \frac{3}{4}\pi$ to limit the risk of damaging the system if the horizontal joint rotates too much. As regards model learning, we considered three different prior definitions to quantify the advantages coming from exploitation of prior information. The kernel considered are: (i) the Gaussian kernel (G), which is the standard option when no prior knowledge is available; (ii) the Gaussian kernel plus a polynomial kernel of degree 2 (G+P$^{(2)}$), which aims at exploiting eventual polynomial behaviors affecting the system dynamics [248]. (iii) semi-parametrical kernel (SP) which combines prior information from physical models and data driven kernels, see [249], [46]:

$$\begin{aligned} k_{SP}(\tilde{x}_{t_j}, \tilde{x}_{t_k}) &= k_{PI}(\tilde{x}_{t_j}, \tilde{x}_{t_k}) + k_G(\tilde{x}_{t_j}, \tilde{x}_{t_k}) \\ &= \phi^T(\tilde{x}_{t_j}) \Sigma_{PI} \phi(\tilde{x}_{t_k}), + k_G(\tilde{x}_{t_j}, \tilde{x}_{t_k}). \end{aligned} \tag{93}$$

where k_{PI} is called a Physically Inspired kernel because it is a linear kernel defined on suitable basis functions $\phi(\tilde{x})$, extracted by first-principles dynamical models, see for instance [249], and Σ_{PI} is a positive-definite matrix. Specifically, SP basis functions can be obtained by isolating, in each ODE defining FP laws of motion, all the linearly related state-dependent components. In particular, we have $\phi_{\dot{\theta}^h}(x,u) = [(\dot{\theta}^v)^2 sin(\theta^v), \dot{\theta}^h \dot{\theta}^v sin(2\theta^v), \dot{\theta}^h, u]$ for the arm velocity GP, and $\phi_{\dot{\theta}^v}(x,u) = [(\dot{\theta}^h)^2 sin(2\theta^v), \dot{\theta}^v, sin(\theta^v), u\ cos(\theta^v)]$ for the pendulum velocity GP.

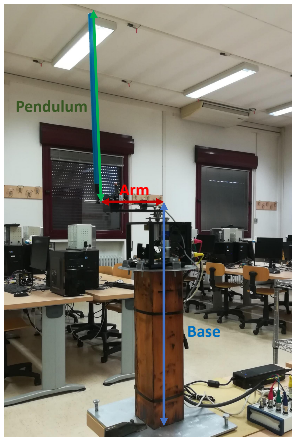

Figure 14 Illustration of the real system Furuta Pendulum controlled in the unstable equilibrium point.

Figure 15 shows the resulting trajectories for each trial. The algorithm learned how to swing up the FP with all the prior models considered. It succeeded at trial 6 with the Gaussian kernel, at trial 4 with kernel G+P$^{(2)}$, and at trial 3 with SP kernel. This result suggests that a great advantage of particle-based approaches is the possibility of using any kernel function, and in particular including prior knowledge on the system dynamics to improve data efficiency.

CONCLUSIONS

In this paper, we have provided an overview of recent advances in kernel-based identification of dynamical systems and their application to control. In the first part of the paper we have reviewed kernel-based methods for linear and nonlinear systems, highlighting the different perspectives and advantages with respect to the classic parametric-based system identification. Looking at kernel-based methods from a Bayesian point of view, we have illustrated the existing bridge between such techniques and Gaussian Process regression, which has been successfully applied in the last decade in different fields, ranging from computer science, data analysis, robotics, and control. Indeed, as discussed in the second part of the paper, GP models allow quantifying the uncertainty of the estimates in a simple and effective way compared to their deterministic counterparts. This makes GPs particularly appealing for model-based control methods since a correct understanding of the uncertainty allows the derivation of more robust control algorithms. In the last part of the paper we have reviewed the use of GPs in control algorithms, such as MPC, feedback linearization, adaptive and robust control, and RL. We have focused on GP-based MBRL

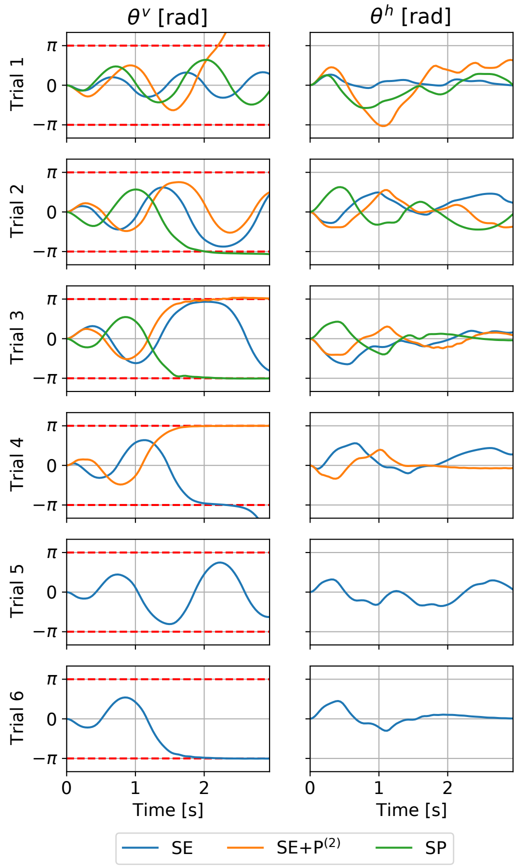

Figure 15 Trajectories of the FP vertical and horizontal angles obtained applying MC-PILCO with prior defined by: G, $G + P^{(2)}$, or SP kernel.

algorithms, a class of algorithms whose aim is to simulate the system evolution and optimize a control policy. We have compared three GP-based MBRL algorithms (PILCO, Black-Drops, and MC-PILCO) on the cart-pole swing-up task, then applying MC-PILCO on a real Furuta pendulum system.

References

[1] L. Zadeh, "On the identification problem," *IRE Transactions on Circuits Theory*, vol. 3, no. 4, pp. 277–281, 1956.

[2] L. Ljung, *System Identification - Theory for the User*, 2nd ed. Upper Saddle River, N.J.: Prentice-Hall, 1999.

[3] T. Söderström and P. Stoica, *System Identification*. Prentice-Hall, 1989.

[4] K. Astrom and T. Bohlin, "Numerical identification of linear dynamic systems from normal operating records," in *Proc. of IFAC Symposium on self-adaptive control systems*, 1965.

[5] G. Casella and R. Berger, *Statistical Inference*. Duxbury Resource Center, June 2001.

[6] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. AC-19, pp. 716–723, 1974.

[7] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, pp. 461–464, 1978.

[8] J. Rissanen, "Modelling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[9] P. D. Grünwald, *The minimum description length principle*. MIT press, 2007.

[10] M. Stone, "Asymptotics for and against cross-validation," *Biometrica*, vol. 64, no. 1, 1977.

[11] T. J. Hastie, R. J. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference and Prediction*. Canada: Springer, 2001.

[12] W. Rudin, *Real and Complex Analysis*. Singapore: McGraw-Hill, 1987.

[13] M. Bertero, "Linear inverse and ill-posed problems," *Advances in Electronics and Electron Physics*, vol. 75, pp. 1–120, 1989.

[14] E. Sontag, "Smooth stabilization implies coprime factorization," *IEEE Transactions on Automatic Control*, vol. 34, no. 4, pp. 435–443, 1989.

[15] W. Lohmiller and J. Slotine, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, pp. 683–696, 1998.

[16] Z. Jing and Y. Wuang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 6, pp. 857 – 869, 2001.

[17] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, ser. (Adaptive Computation and Machine Learning). MIT Press, 2001.

[18] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. Mc Graw-Hill, 1984.

[19] N. Aronszajn, "Theory of reproducing kernels," *Trans. of the American Mathematical Society*, vol. 68, pp. 337–404, 1950.

[20] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, March 2014.

[21] G. Pillonetto and G. De Nicolao, "A new kernel-based approach for linear system identification," *Automatica*, vol. 46, no. 1, pp. 81–93, January 2010.

[22] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[23] A. Y. Aravkin, B. M. Bell, J. V. Burke, and G. Pillonetto, "The connection between bayesian estimation of a gaussian random field and rkhs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1518–1524, 2015.

[24] G. Kimeldorf and G. Wahba, "A correspondence between Bayesian estimation on stochastic processes and smoothing by splines," *The Annals of Mathematical Statistics*, vol. 41, no. 2, pp. 495–502, 1970.

[25] G. Pillonetto, T. Chen, A. Chiuso, G. De Nicolao, and L. Ljung, "Regularized linear system identification using atomic, nuclear and kernel-based norms: The role of the stability constraint," *Automatica*, vol. 69, pp. 137 – 149, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109816300449

[26] T. Chen, M. S. Andersen, L. Ljung, A. Chiuso, and G. Pillonetto, "System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2933–2945, 2014.

[27] T. Chen, "On kernel design for regularized LTI system identification," *Automatica*, vol. 90, pp. 109–122, 2018.

[28] M. Darwish, G. Pillonetto, and R. Toth, "The quest for the right kernel in Bayesian impulse response identification: The use of OBFs," *Automatica*, vol. 87, pp. 318 – 329, 2018.

[29] G. Pillonetto, T. Chen, A. Chiuso, G. D. Nicolao, and L. Ljung, *Regularized System Identification*. Springer, 2022.

[30] J. Lataire and T. Chen, "Transfer function and transient estimation by gaussian process regression in the frequency domain," *Automatica*, vol. 72, pp. 217–229, 2016.

[31] M. Zorzi and A. Chiuso, "The harmonic analysis of kernel functions," *Automatica*, vol. 94, pp. 125–137, 2018.

[32] E. Jaynes, "On the rationale of maximum-entropy methods," *Proceedings of IEEE*, vol. 70, pp. 939–952, 1982.

[33] T. Chen, T. Ardeshiri, F. P. Carli, A. Chiuso, L. Ljung, and G. Pillonetto, "Maximum entropy properties of discrete-time first-order stable spline kernel," *Automatica*, vol. 66, pp. 34 – 38, 2016.

[34] F. P. Carli, T. Chen, and L. Ljung, "Maximum entropy kernels for system identification," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp.

1471–1477, 2017.

[35] F. Dinuzzo, "Kernels for linear time invariant system identification," *SIAM Journal on Control and Optimization*, vol. 53, no. 5, pp. 3299–3317, 2015.

[36] M. Bisiacco and G. Pillonetto, "Kernel absolute summability is sufficient but not necessary for rkhs stability," *SIAM journal on control and optimization*, 2020.

[37] ——, "On the mathematical foundations of stable rkhss," *Automatica*, 2020.

[38] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series," *IEEE Transactions on Circuits and Systems*, vol. 32, no. 11, pp. 1150–1161, 1985.

[39] C. Cheng, Z. Peng, W. Zhang, and G. Meng, "Volterra-series-based nonlinear system modeling and its engineering applications: A state-of-the-art review," *Mechanical Systems and Signal Processing*, vol. 87, pp. 340 – 364, 2017.

[40] M. Franz and B. Schölkopf, "A unifying view of Wiener and Volterra theory and polynomial kernel regression," *Neural Computation*, vol. 18, pp. 3097–3118, 2006.

[41] G. Birpoutsoukis, A. Marconato, J. Lataire, and J. Schoukens, "Regularized nonparametric volterra kernel estimation," *Automatica*, vol. 82, pp. 324 – 327, 2017.

[42] J. G. Stoddard, J. S. Welsh, and H. Hjalmarsson, "EM-based hyperparameter optimization for regularized volterra kernel estimation," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 388–393, 2017.

[43] A. Dalla Libera, R. Carli, and G. Pillonetto, "Kernel-based methods for Volterra series identification," *Automatica*, vol. 129, p. 109686, 2021.

[44] A. Dalla Libera, R. Carli, and G. Pillonetto, "A novel multiplicative polynomial kernel for volterra series identification," in *2020 World Congress of the International Federation of Automatic Control (IFAC)*, 2020.

[45] D. Romeres, M. Zorzi, R. Camoriano, and A. Chiuso, "Online semi-parametric learning for inverse dynamics modeling," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 2945–2950.

[46] D. Nguyen-Tuong and J. Peters, "Using model knowledge for learning inverse dynamics," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2677–2682.

[47] D. Romeres, M. Zorzi, R. Camoriano, S. Traversaro, and A. Chiuso, "Derivative-free online learning of inverse dynamics models," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 816–830, 2019.

[48] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth, "Manifold gaussian processes for regression," in *2016 International joint conference on neural networks (IJCNN)*. IEEE, 2016, pp. 3338–3345.

[49] A. Dalla Libera and R. Carli, "A data-efficient geometrically inspired polynomial kernel for robot inverse dynamic," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 24–31, 2019.

[50] A. Dalla Libera, D. Romeres, D. K. Jha, B. Yerazunis, and D. Nikovski, "Model-based reinforcement learning for physical systems without velocity and acceleration measurements," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3548–3555, 2020.

[51] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, *Bounding approaches to system identification*. Springer Science & Business Media, 2013.

[52] G. Pillonetto, T. Chen, A. Chiuso, G. De Nicolao, and L. Ljung, *Bayesian Interpretation of Regularization*. Regularized System Identification: Learning Dynamic Models from Data: Springer International Publishing, 2022, pp. 95–134.

[53] G. Casella, "An introduction to empirical Bayes data analysis," *The American Statistician*, vol. 39, no. 2, pp. 83–87, 1985.

[54] D. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, pp. 415–447, 1992.

[55] F. Berkenkamp and A. P. Schoellig, "Safe and robust learning control with gaussian processes," in *Proceedings of European Control Conference (ECC)*, 2015, pp. 2496–2501.

[56] A. Scampicchio, A. Chiuso, S. Formentin, and G. Pillonetto, "Bayesian kernel-based linear control design," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 822–827.

[57] A. Scampicchio, A. Aravkin, and G. Pillonetto, "Stable and robust lqr design via scenario approach," *Automatica*, vol. 129, 2021.

[58] R. C. Grande, G. Chowdhary, and J. P. How, "Experimental validation of bayesian nonparametric adaptive control using gaussian processes," *Journal of Aerospace Information Systems*, vol. 11, pp. 565–578, 2014.

[59] G. Chowdhary, H. A. Kingravi, and J. P. How, "Bayesian nonparametric adaptive control using gaussian processes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 537–550, 2015.

[60] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche, "Feedback linearization using gaussian processes," in *IEEE Conference on Decision and Control (CDC)*, 2017, pp. 5249–5255.

[61] M. Helwa, A. Heins, and A. P. Schoellig, "Provably robust learning-based approach for high-accuracy tracking control of lagrangian systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1587–1594, 2010.

[62] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2020.

[63] Y. Wang, C. Ocampo-Martinez, and V. Puig, "Stochastic model predictive control based on gaussian processes applied to drinking water networks," *IET Control Theory and Applications*, vol. 10, no. 8, pp. 947–955, 2016.

[64] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[65] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*. Citeseer, 2011, pp. 465–472.

[66] M. Cutler and J. P. How, "Efficient reinforcement learning for robots using informative simulated priors," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2605–2612.

[67] Y. Gal, R. McAllister, and C. E. Rasmussen, "Improving pilco with bayesian neural network dynamics models," in *Data-Efficient Machine Learning workshop, ICML*, vol. 4, 2016, p. 34.

[68] P. Parmas, C. E. Rasmussen, J. Peters, and K. Doya, "Pipps: Flexible model-based policy search robust to the curse of chaos," in *International Conference on Machine Learning*, 2018, pp. 4065–4074.

[69] F. Amadio, A. Dalla Libera, R. Antonello, D. Nikovski, R. Carli, and D. Romeres, "Model-based policy search using monte carlo gradient estimation with real systems application," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3879–3898, 2022.

[70] K. Chatzilygeroudis, R. Rama, R. Kaushik, D. Goepp, V. Vassiliades, and J. Mouret, "Black-box data-efficient policy search for robotics," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 51–58.

[71] A. Hoerl, "Application of ridge analysis to regression problems," *Chemical Engineering Progress*, vol. 58, pp. 54–59, 1962.

[72] A. Hoerl and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55–67, 1970.

[73] G. Wahba, *Spline models for observational data*. Philadelphia: SIAM, 1990.

[74] B. Bell and G. Pillonetto, "Estimating parameters and stochastic functions of one variable using nonlinear measurement models," *Inverse Problems*, vol. 20, no. 3, p. 627, 2004.

[75] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," *Neural Networks and Computational Learning Theory*, vol. 81, pp. 416–426, 2001.

[76] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, N.J., USA: Prentice-Hall, 1979.

[77] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. IOP Publishing, 1998.

[78] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. Philadelphia: SIAM, 2005.

[79] D. Phillips, "A technique for the numerical solution of certain integral equations of the first kind," *J Assoc Comput Mach*, vol. 9, pp. 84–97, 1962.

[80] O. Toeplitz, "Zur theorie der quadratischen und bilinearen formen von unendlichvielen veränderlichen," *Mathematische Annalen*, vol. 70, no. 3, pp. 351–376, 1911.

[81] U. Grenander and G. Szegö, *Toeplitz forms and their applications*. University of California Press, 1956, vol. 321.

[82] B. Wahlberg, "System identification using Laguerre models," *IEEE Trans. Automatic Control*, vol. AC-36, pp. 551–562, 1991.

[83] T. Chen, H. Ohlsson, and L. Ljung, "On the estimation of transfer functions, regularizations and Gaussian processes - Revisited," *Automatica*,

vol. 48, pp. 1525–1535, 2012.

[84] G. Pillonetto, A. Chiuso, and G. De Nicolao, "Regularized estimation of sums of exponentials in spaces generated by stable spline kernels," in *Proceedings of the IEEE American Cont. Conf., Baltimora, USA*, 2010.

[85] G. Prando, D. Romeres, G. Pillonetto, and A. Chiuso, "Classical vs. bayesian methods for linear system identification: Point estimators and confidence sets," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 1365–1370.

[86] V. L. Mehrmann, *The autonomous linear quadratic control problem*, 1991st ed., ser. Lecture Notes in Control and Information Sciences. Berlin, Germany: Springer, Oct. 1991.

[87] 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, no. 4, pp. 633–679, 2020.

[88] A. Lindquist, "On feedback control of linear stochastic systems," *SIAM Journal on Control*, vol. 11, no. 2, pp. 323–343, 1973. [Online]. Available: https://doi.org/10.1137/0311025

[89] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference, ACC 2005, Portland, OR, USA, 8-10 June, 2005*. IEEE, 2005, pp. 300–306vol.1. [Online]. Available: https://doi.org/10.1109/ACC.2005.1469949

[90] A. Varga, "Balancing free square-root algorithm for computing singular perturbation approximations," in *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, 1991, pp. 1062–1065 vol.2.

[91] M. Green, "A relative error bound for balanced stochastic truncation," *IEEE Transactions on Automatic Control*, vol. 33, no. 10, pp. 961–965, 1988.

[92] G. Pillonetto, A. Chiuso, and G. De Nicolao, "Prediction error identification of linear systems: a nonparametric Gaussian regression approach," *Automatica*, vol. 47, no. 2, pp. 291–305, 2011.

[93] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, May 1979.

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

[95] B. Efron and C. Morris, "Stein's estimation rule and its competitors–an empirical Bayes approach," *Journal of the American Statistical Association*, vol. 68, no. 341, pp. 117–130, 1973.

[96] A. Chiuso, "Regularization and Bayesian learning in dynamical systems: Past, present and future," *Annual Reviews in Control*, vol. 41, pp. 24 – 38, 2016.

[97] A. Aravkin, J. V. Burke, A. Chiuso, and G. Pillonetto, "Convex vs nonconvex estimators for regression and sparse estimation: the mean squared error properties of ard and glasso," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 217–252, 2014.

[98] G. Pillonetto and A. Chiuso, "Tuning complexity in regularized kernel-based regression and linear system identification: The robustness of the marginal likelihood estimator," *Automatica*, vol. 51, pp. 106 – 117, 2015.

[99] D. Romeres, G. Prando, G. Pillonetto, and A. Chiuso, "On-line bayesian system identification," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 1359–1364.

[100] G. Prando, D. Romeres, and A. Chiuso, "Online identification of time-varying systems: A bayesian approach," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 3775–3780.

[101] S. Bergman, *The Kernel Function and Conformal Mapping*. Mathematical Surveys and Monographs, AMS, 1950.

[102] A. Tikhonov, "On the solution of incorrectly formulated problems and the regularization method," *Doklady Akademii Nauk SSSR*, vol. 151, pp. 501–504, 1963.

[103] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*. Washington, D.C.: Winston/Wiley, 1977.

[104] M. Bertero, T. Poggio, and V. Torre, "Ill-posed problems in early vision," *Proceedings of IEEE*, 1988.

[105] T. Poggio and F. Girosi, "Networks for approximation and learning," in *Proceedings of the IEEE*, vol. 78, 1990, pp. 1481–1497.

[106] F. Girosi, "An equivalence between sparse approximation and support vector machines," *Neural Comput.*, vol. 10, no. 6, pp. 1455–1480, Aug. 1998.

[107] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[108] J. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[109] E. E. Leamer, "A class of informative priors and distributed lag analysis," *Econometrica*, vol. 40, no. 6, pp. pp. 1059–1081, 1972.

[110] R. J. Shiller, "A distributed lag estimator derived from smoothness priors," *Econometrica*, vol. 41, no. 4, pp. pp. 775–788, 1973.

[111] H. Akaike, "Smoothness priors and the distributed lag estimator," Department of Statistics, Stanford University, Tech. Rep., 1979.

[112] G. Kitagawa and W. Gersch, *Smoothness Priors Analysis of Time Series*, ser. Ima Volumes in Mathematics and Its Applications. Springer New York, 1996.

[113] G. Goodwin, M. Gevers, and B. Ninness, "Quantifying the error in estimated transfer functions with application to model order selection," *IEEE Transactions on Automatic Control*, vol. 37, no. 7, pp. 913–928, 1992.

[114] R. M. Neal, *Bayesian learning for neural networks*, 1995, vol. 118.

[115] C. Williams, "Computation with infinite neural networks," *Neural Computation*, vol. 10, no. 5, pp. 1203–1216, 1998.

[116] J. Lee, Y. Bahri, R. Novak, S. Schoenholz, J. Pennington, and J. Sohldickstein, "Deep neural networks as Gaussian processes," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1EA-M-0Z

[117] R. Novak, L. Xiao, J. Lee, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Bayesian Convolutional Neural Networks with Many Channels are Gaussian Processes," *arXiv:1810.05148 [cs, stat]*, Oct. 2018, arXiv: 1810.05148. [Online]. Available: http://arxiv.org/abs/1810.05148

[118] G. Yang, "Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/5e69fda38cda2060819766569fd93aa5-Abstract.html

[119] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in Computational Mathematics*, vol. 13, pp. 1–50, 2000.

[120] A. Argyriou and F. Dinuzzo, "A unifying view of representer theorems," in *Proceedings of the 31th International Conference on Machine Learning*, vol. 32, 2014, pp. 748–756.

[121] C. Cobelli and E. Carson, *Introduction to modeling in physiology and medicine*. Academic Press, 2019.

[122] P. van den Driesschea and J. Watmoughb, "Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission," *Mathematical Biosciences*, vol. 180, pp. 29 – 48, 2002.

[123] J. Schoukens and L. Ljung, "Nonlinear system identification – a user-oriented roadmap," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, December 2019.

[124] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley, 1980.

[125] S. Chen, C. F. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neur. Netw.*, vol. 2, no. 2, pp. 302–309, 1991.

[126] S. Billings and W. Hua-Liang, "A new class of wavelet networks for nonlinear system identification," *IEEE Transactions on Neural Networks*, vol. 16, pp. 862 – 874, 2005.

[127] X. Hong, R. J. Mitchell, S. Chen, C. J. Harris, K. Li, and G. W. Irwin, "Model selection approaches for non-linear system identification: A review," *International Journal of Systems Science*, vol. 39, no. 10, pp. 925–946, 2008.

[128] I. Lind and L. Ljung, "Regressor selection with the analysis of variance method," *Automatica*, vol. 41, no. 4, pp. 693 – 700, 2005.

[129] ——, "Regressor and structure selection in NARX models using a structured ANOVA approach," *Automatica*, vol. 44, pp. 383–395, 2008.

[130] P. Huber, "Projection pursuit," *Ann. Statist*, vol. 13, pp. 435–475, 1985.

[131] H. Ohlsson, J. Roll, T. Glad, and L. Ljung, "Using manifold learning for nonlinear system identification," in *Proc IFAC Symposium on Nonlinear Conrtrol Systems (NOLCOS*. Pretoria, South Africa: IFAC, August 2007.

[132] S. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol. 290, no. 2323-2326, 2000.

[133] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces," *Journal of*

Machine Learning Research, vol. 5, pp. 73–99, 2004.

[134] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 58, pp. 267–288, 1996.

[135] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[136] L. Rosasco, S. Villa, S. Mosci, M. Santoro, and A. Verri, "Nonparametric sparsity and regularization," *Journal of Machine Learning Research*, vol. 14, pp. 1665–1714, 2013.

[137] B. Mu, W. X. Zheng, and E. Bai, "Variable selection and identification of high-dimensional nonparametric additive nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2254–2269, 2017.

[138] C. Cheng, E. Bai, and Z. Peng, "Consistent variable selection for a nonparametric nonlinear system by inverse and contour regressions," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2653–2664, 2019.

[139] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1235–1256, 2009.

[140] K. Mohan and M. Fazel, "Reweighted nuclear norm minimization with application to system identification," in *American Control Conference (ACC)*, 2010, pp. 2953–2959.

[141] V. Chandrasekaran, B. Recht, P. Parrilo, and A. Willsky, "The convex geometry of linear inverse problems," *Foundations of Computational Mathematics*, vol. 12, no. 6, pp. 805–849, 2012.

[142] R. Smith, "Frequency domain subspace identification using nuclear norm minimization and Hankel matrix realizations," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2886–2896, 2014.

[143] L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds., *Large Scale Kernel Machines*. Cambridge, MA, USA: MIT Press, 2007.

[144] K. Bennett and E. Parrado-Hernandez, "The interplay of optimization and machine learning research," *J. of Machine Learning Research*, vol. 7, pp. 1265–1281, 2006.

[145] R. Rockafellar, *Convex Analysis*, ser. Princeton Landmarks in Mathematics. Princeton University Press, 1970.

[146] F. Bach and M. Jordan, "Predictive low-rank decomposition for kernel methods," in *Proceedings of the 22nd international conference on Machine learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 33–40.

[147] B. Kulis, M. Sustik, and I. Dhillon, "Learning low-rank kernel matrices," in *Proceedings of the 23rd international conference on Machine learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 505–512.

[148] C. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proceedings of the 2000 conference on Advances in neural information processing systems*. Cambridge, MA, USA: MIT Press, 2000, pp. 682–688.

[149] K. Zhang and J. Kwok, "Clustered Nyström method for large scale manifold learning and dimension reduction," *IEEE Trans. on Neural Networks*, vol. 21, no. 10, pp. 1576–1587, Oct. 2010.

[150] A. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 911–918.

[151] H. Zhu and R. Rohwer, "Bayesian regression filters and the issue of priors," *Neural Computing and Applications*, vol. 4, pp. 130–142, 1996.

[152] H. Zhu, C. Williams, R. Rohwer, and M. Morciniec, "Gaussian regression and optimal finite dimensional linear models," in *Neural Networks and Machine Learning*. Berlin: Springer-Verlag, 1998.

[153] G. Ferrari-Trecate, C. Williams, and M. Opper, "Finite-dimensional approximation of Gaussian processes," in *Proceedings of the 1998 conference on Advances in neural information processing systems*. Cambridge, MA, USA: MIT Press, 1999, pp. 218–224.

[154] G. Pillonetto and B. Bell, "Bayes and empirical Bayes semi-blind deconvolution using eigenfunctions of a prior covariance," *Automatica*, vol. 43, no. 10, pp. 1698–1712, 2007.

[155] F. Carli, A. Chiuso, and G. Pillonetto, "Efficient algorithms for large scale linear system identification using stable spline estimators," in *IFAC symposium on system identification*, 2012.

[156] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07, 2007, pp. 1177–1184.

[157] A. Rudi and L. Rosasco, "Generalization properties of learning with random features," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, pp. 3218–3228.

[158] H. Liu, Y. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable gps," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4405–4423, 2020.

[159] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *J. of Machine Learning Research*, vol. 7, pp. 2651–2667, December 2006.

[160] M. Espinoza, J. A. K. Suykens, and B. De Moor, "Kernel based partially linear models and nonlinear identification," *IEEE Trans. on Automatic Control*, vol. 50, no. 10, pp. 1602–1606, 2005.

[161] Y. Li, L. Li, H. Su, and J. Chun, "Least squares support vector machine based partially linear model identification," in *Intelligent Computing*, ser. Lecture Notes in Computer Science, D.-S. Huang, K. Li, and G. Irwin, Eds. Springer Berlin Heidelberg, 2006, vol. 4113, pp. 775–781.

[162] Y. Xu and D. Chen, "Partially-linear least-squares regularized regression for system identification." *IEEE Trans. Automat. Contr.*, vol. 54, no. 11, pp. 2637–2641, 2009.

[163] G. Pillonetto, "System identification using kernel-based regularization: New insights on stability and consistency issues," *Automatica*, vol. 93, pp. 321 – 332, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109818301602

[164] R. Frigola, F. Lindsten, T. Schon, and C. Rasmussen, "Bayesian inference and learning in Gaussian process state-space models with particle mcmc," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

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

[166] M. Campi, A. Carè, and S. Garatti, "The scenario approach: A tool at the service of data-driven decision making," *Annual Reviews in Control*, vol. 52, pp. 1–17, 2021.

[167] A. Girard, C. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting," *Advances in neural information processing systems*, vol. 15, 2002.

[168] R. Pintelon and J. Schoukens, *System identification: a frequency domain approach*, 2nd ed. John Wiley & Sons, 2012.

[169] E. Fogel, "System identification via membership set constraints with energy constrained noise," *IEEE Transactions on Automatic Control*, vol. 24, no. 5, pp. 752–758, 1979.

[170] E. Fogel and Y. F. Huang, "On the value of information in system identification—bounded noise case," *Automatica*, vol. 18, no. 2, pp. 229–238, 1982. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109882901108

[171] J. P. Norton, "Identification of parameter bounds for armax models from records with bounded noise," *International Journal of Control*, vol. 45, no. 2, pp. 375–390, 1987.

[172] M. Milanese and A. Vicino, "Optimal estimation theory for dynamic systems with set membership uncertainty: An overview," *Automatica*, vol. 27, no. 6, pp. 997–1009, 1991.

[173] M. Milanese and M. Taragna, "H_∞ set membership identification: A survey," *Automatica*, vol. 41, no. 12, pp. 2019 – 2032, 2005.

[174] M. Milanese and C. Novara, "Unified set membership theory for identification, prediction and filtering of nonlinear systems," *Automatica*, vol. 47, no. 10, pp. 2141 – 2151, 2011.

[175] J. Chen and G. Gu, *Control-oriented system identification: an H_∞ approach*. Wiley-Interscience, 2000, vol. 19.

[176] M. Karimshoushtari and C. Novara, "Design of experiments for nonlinear system identification: A set membership approach," *Automatica*, vol. 119, p. 109036, 2020.

[177] J.-P. Calliess, S. J. Roberts, C. E. Rasmussen, and J. Maciejowski, "Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control," *Automatica*, vol. 122, p. 109216, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109820304143

[178] F. Dabbene, M. Sznaier, and R. Tempo, "Probabilistic optimal estimation with uniformly distributed noise," *IEEE Transactions on Automatic Control*,

vol. 59, no. 8, pp. 2113–2127, 2014.

[179] G. Baggio, A. Carè, A. Scampicchio, and G. Pillonetto, "Bayesian frequentist bounds with applications to machine learning and system identification," *Automatica*, 2022, to appear.

[180] S. Garatti, M. C. Campi, and S. Bittanti, "Assessing the quality of identified models through the asymptotic theory—when is the result reliable?" *Automatica*, vol. 40, no. 8, pp. 1319–1332, 2004.

[181] H. Leeb and B. M. Pötscher, "Model selection and inference: Facts and fiction," *Econometric Theory*, vol. 21, no. 1, pp. 21–59, 2005.

[182] B. Cs. Csáji, M. C. Campi, and E. Weyer, "Sign-perturbed sums: A new system identification approach for constructing exact non-asymptotic confidence regions in linear regression models," *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 169–181, 2015.

[183] E. Weyer, M. C. Campi, and B. Cs. Csáji, "Asymptotic properties of SPS confidence regions," *Automatica*, vol. 82, pp. 287 – 294, 2017.

[184] A. Carè, B. Cs. Csáji, M. C. Campi, and E. Weyer, "Finite-sample system identification: An overview and a new correlation method," *IEEE Control Systems Letters*, vol. 2, no. 1, pp. 61–66, Jan 2018.

[185] M. C. Campi and E. Weyer, "Guaranteed non-asymptotic confidence regions in system identification," *Automatica*, vol. 41, no. 10, pp. 1751 – 1764, 2005.

[186] M. Dalai, E. Weyer, and M. C. Campi, "Parameter identification for nonlinear systems: guaranteed confidence regions through LSCR," *Automatica*, vol. 43, pp. 1418–1425, 2007.

[187] O. N. Granichin, "The nonasymptotic confidence set for parameters of a linear control object under an arbitrary external disturbance," *Automation and Remote Control*, vol. 73, no. 1, pp. 20–30, 2012.

[188] S. Kolumbán, I. Vajk, and J. Schoukens, "Perturbed datasets methods for hypothesis testing and structure of corresponding confidence sets," *Automatica*, vol. 51, pp. 326–331, 2015.

[189] K. Amelin and O. Granichin, "Randomized control strategies under arbitrary external noise," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1328–1333, 2016.

[190] M. V. Volkova, O. N. Granichin, Y. V. Petrov, and G. A. Volkov, "Dynamic fracture tests data analysis based on the randomized approach," *Advances in Systems Science and Applications*, vol. 17, no. 3, pp. 34–41, 2017.

[191] C.-Y. Han, M. Kieffer, and A. Lambert, "Guaranteed confidence region characterization for source localization using RSS measurements," *Signal Processing*, vol. 152, pp. 104–117, 2018.

[192] B. C. Csáji and K. B. Kis, "Distribution-free uncertainty quantification for kernel methods by gradient perturbations," *Machine Learning*, vol. 108, no. 8, pp. 1677–1699, Sep 2019. [Online]. Available: https://doi.org/10.1007/s10994-019-05822-1

[193] M. M. Khorasani and E. Weyer, "Non-asymptotic confidence regions for the parameters of eiv systems," *Automatica*, vol. 115, p. 108873, 2020.

[194] ——, "Non-asymptotic confidence regions for the transfer functions of errors-in-variables systems," *IEEE Transactions on Automatic Control*, pp. 1–1, 2021.

[195] G. Baggio, A. Carè, and G. Pillonetto, "Finite-sample guarantees for state-space system identification under full state measurements," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 2789–2794.

[196] A. Carè, "A simple condition for the boundedness of sign-perturbed-sums (SPS) confidence regions," *Automatica*, vol. 139, p. 110150, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109821006798

[197] A. Goldenshluger, "Nonparametric estimation of transfer functions: rates of convergence and adaptation," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 644–658, 1998.

[198] E. Weyer, R. C. Williamson, and I. M. Mareels, "Finite sample properties of linear model identification," *IEEE Transactions on Automatic Control*, vol. 44, no. 7, pp. 1370–1383, 1999.

[199] M. C. Campi and E. Weyer, "Finite sample properties of system identification methods," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1329–1334, 2002.

[200] M. Vidyasagar and R. L. Karandikar, "A learning theory approach to system identification and stochastic adaptive control," *Probabilistic and randomized methods for design under uncertainty*, pp. 265–302, 2006.

[201] Y. Abbasi-Yadkori and C. Szepesvári, "Regret bounds for the adaptive control of linear quadratic systems," in *Proceedings of the 24th Annual Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 2011, pp. 1–26.

[202] J. Pereira, M. Ibrahimi, and A. Montanari, "Learning networks of stochastic differential equations," *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[203] P. Shah, B. N. Bhaskar, G. Tang, and B. Recht, "Linear system identification via atomic norm regularization," in *2012 IEEE 51st IEEE conference on decision and control (CDC)*. IEEE, 2012, pp. 6265–6270.

[204] R. Boczar, N. Matni, and B. Recht, "Finite-data performance guarantees for the output-feedback control of an unknown system," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 2994–2999.

[205] M. Hardt, T. Ma, and B. Recht, "Gradient descent learns linear dynamical systems," *Journal of Machine Learning Research*, vol. 19, pp. 1–44, 2018.

[206] S. Oymak and N. Ozay, "Non-asymptotic identification of LTI systems from a single trajectory," in *2019 American control conference (ACC)*. IEEE, 2019, pp. 5655–5661.

[207] A. Tsiamis and G. J. Pappas, "Finite sample analysis of stochastic system identification," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 3648–3654.

[208] S. Oymak and N. Ozay, "Revisiting Ho–Kalman-based system identification: Robustness and finite-sample analysis," *IEEE Transactions on Automatic Control*, vol. 67, no. 4, pp. 1914–1928, 2021.

[209] A. Tsiamis and G. J. Pappas, "Linear systems can be hard to learn," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 2903–2910.

[210] T. Sarkar, A. Rakhlin, and M. A. Dahleh, "Finite time LTI system identification," 2021.

[211] B. C. Csáji, "Non-asymptotic confidence regions for regularized linear regression estimates," in *Progress in Industrial Mathematics at ECMI 2018*, I. Faragó, F. Izsák, and P. L. Simon, Eds. Cham: Springer International Publishing, 2019, pp. 605–611.

[212] V. Volpe, *Identification of dynamical systems with finitely many data points*. University of Brescia, M. Sc. Thesis, March 2015.

[213] A. Carè, G. Pillonetto, and M. C. Campi, "Uncertainty bounds for kernel-based regression: A bayesian sps approach," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2018, pp. 1–6.

[214] G. Pillonetto, A. Carè, and M. C. Campi, "Kernel-based sps," *IFAC-PapersOnLine*, vol. 51, no. 15, pp. 31 – 36, 2018, 18th IFAC Symposium on System Identification SYSID 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896318317464

[215] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules," *IEEE Signal Processing Magazine*, vol. 21, no. 4, pp. 36–47, 2004.

[216] L. Ljung, "Model validation and model error modeling," Linköping University, Automatic Control, Tech. Rep. 2125, 1999.

[217] A. Carè, M. C. Campi, B. r. Csáji, and E. Weyer, "Facing undermodelling in Sign-Perturbed-Sums system identification," *Systems & Control Letters*, vol. 153, p. 104936, 2021.

[218] G. C. Goodwin and M. E. Salgado, "A stochastic embedding approach for quantifying uncertainty in the estimation of restricted complexity models," *International Journal of Adaptive Control and Signal Processing*, vol. 3, no. 4, pp. 333–356, 1989.

[219] G. C. Goodwin, M. Gevers, and D. Q. Mayne, "Bias and variance distribution in transfer function estimation," *IFAC Proceedings Volumes*, vol. 24, no. 3, pp. 811–816, 1991, 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation 1991, Budapest, Hungary, 8-12 July 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017524490

[220] G. C. Goodwin, J. H. Braslavsky, and M. M. Seron, "Non-stationary stochastic embedding for transfer function estimation," *Automatica*, vol. 38, no. 1, pp. 47–62, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109801001868

[221] L. Ljung, G. C. Goodwin, J. C. Agüero, and T. Chen, "Model error modeling and stochastic embedding," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 75–79, 2015, 17th IFAC Symposium on System Identification SYSID 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896315027275

[222] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis, third ed.* New York, NY: Chapman and Hall/CRC, 2013.

[223] J. Kocijan and R. Murray-Smith, "Nonlinear predictive control with a gaussian process model," *Lecture Notes in Computer Science*, vol. 3355, pp. 185–200, 2005.

[224] K. Ota, D. K. Jha, D. Romeres, J. van Baar, K. A. Smith, T. Semitsu, T. Oiki, A. Sullivan, D. Nikovski, and J. B. Tenenbaum, "Data-efficient learning for complex and real-time physical problem solving using augmented simulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4241–4248, 2021.

[225] S. Kamthe and M. Deisenroth, "Data-efficient reinforcement learning with probabilistic model predictive control," in *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, 2018.

[226] T. Koller, F. Berkenkamp, M. Turchetta, J. Boedecker, and A. Krause, "Learning-based model predictive control for safe exploration and reinforcement learning," *arXiv:1906.12189*, 2019.

[227] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based nmpc enabling reliable mobile robot path tracking," *The International Journal of Robotics Research*, vol. 35, pp. 1547–1563, 2016.

[228] E. Picotti, A. Dalla Libera, R. Carli, and M. Bruschetta, "Lbmatmpc: an open-source toolbox for gaussian process modeling within learning-based nonlinear model predictive controls," in *European Control Conference (ECC)*, 2022.

[229] A. Dalla Libera, F. Amadio, D. Nikovski, R. Carli, and D. Romeres, "Control of mechanical systems via feedback linearization based on black-box gaussian process models," in *2021 European Control Conference (ECC)*, 2021.

[230] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*, ser. Dover Books on Electrical Engineering Series. Dover Publications, 2011.

[231] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, pp. 319–340, 2011.

[232] G. Joshi and G. Chowdhary, "Adaptive control using gaussian-process with model reference generative network," in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 237–243.

[233] G. Dullerud and F. Paganini, *A Course in Robust Control Theory*, ser. Texts in Applied Mathematics. Springer, 2000.

[234] Y. Engel, S. Mannor, and R. Meir, "Bayes meets bellman: The gaussian process approach to temporal difference learning," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 154–161.

[235] ——, "Reinforcement learning with gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 201–208.

[236] R. Grande, T. Walsh, and J. How, "Sample efficient reinforcement learning with gaussian processes," in *International Conference on Machine Learning*. PMLR, 2014, pp. 1332–1340.

[237] Y. Fan, L. Chen, and Y. Wang, "Efficient model-free reinforcement learning using gaussian process," *arXiv preprint arXiv:1812.04359*, 2018.

[238] V. Suryan, N. Gondhalekar, and P. Tokekar, "Multifidelity reinforcement learning with gaussian processes: model-based and model-free algorithms," *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 117–128, 2020.

[239] S. Bansal, R. Calandra, K. Chua, S. Levine, and C. Tomlin, "Mbmf: Model-based priors for model-free reinforcement learning," *arXiv preprint arXiv:1709.03153*, 2017.

[240] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *ACM Sigart Bulletin*, vol. 2, no. 4, pp. 160–163, 1991.

[241] A.-m. Farahmand, A. Barreto, and D. Nikovski, "Value-aware loss function for model-based reinforcement learning," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1486–1494.

[242] A.-m. Farahmand, "Iterative value-aware model learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[243] M. P. Deisenroth, C. E. Rasmussen, and D. Fox, "Learning to control a low-cost manipulator using data-efficient reinforcement learning," *Robotics: Science and Systems VII*, pp. 57–64, 2011.

[244] M. P. Deisenroth, R. Calandra, A. Seyfarth, and J. Peters, "Toward fast policy search for learning legged locomotion," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1787–1792.

[245] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Advances in Neural Information Processing Systems*, 2018, pp. 4754–4765.

[246] A. J. McHutchon *et al.*, "Nonlinear modelling and control using gaussian processes," Ph.D. dissertation, Citeseer, 2015.

[247] F. Amadio, A. D. Libera, D. Nikovski, R. Carli, and D. Romeres, "Learning control from raw position measurements," *arXiv preprint arXiv:2301.13183*, 2023.

[248] A. D. Libera and R. Carli, "A data-efficient geometrically inspired polynomial kernel for robot inverse dynamic," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 24–31, 2020.

[249] D. Romeres, D. K. Jha, A. DallaLibera, B. Yerazunis, and D. Nikovski, "Semiparametrical gaussian processes learning of forward dynamical models for navigating in a circular maze," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3195–3202.

[250] D. J. MacKay, "Bayesian methods for adaptive models," Ph.D. dissertation, California Institute of Technology, 1992.

[251] A. Y. Ng and M. I. Jordan, "Pegasus: A policy search method for large mdps and pomdps," *arXiv preprint arXiv:1301.3878*, 2013.

[252] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[253] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International conference on machine learning*. PMLR, 2014, pp. 1278–1286.

[254] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.

[255] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[256] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[257] B. S. Cazzolato and Z. Prime, "On the dynamics of the furuta pendulum," *Journal of Control Science and Engineering*, vol. 2011, 2011.

AUTHOR BIOGRAPHY

Algo Carè (Member, IEEE) received the Ph.D. degree in informatics and automation engineering in 2013 from the University of Brescia, Italy, where he is currently a Research Fellow with the Department of Information Engineering. After his Ph.D., he spent two years at the University of Melbourne, VIC, Australia, as a Research Fellow in system identification with the Department of Electrical and Electronic Engineering. In 2016, he was a recipient of a two-year ERCIM Fellowship that he spent at the Institute for Computer Science and Control (SZTAKI), Hungarian Academy of Sciences (MTA), Budapest, Hungary, and with the Multiscale Dynamics Group, National Research Institute for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands. He received the triennial Stochastic Programming Student Paper Prize by the Stochastic Programming Society for the period 2013–2016. He is associate editor of the International Journal of Adaptive Control and Signal Processing, member of the EUCA Conference Editorial Board, of the IFAC Technical Committee on Modeling, Identification and Signal Processing, and of the IEEE Technical Committee on Systems Identification and Adaptive Control. His current research interests include data-driven decision methods, system identification, and learning theory.

Picture taken at Keukenhof, the Netherlands.

Ruggero Carli received the Laurea degree in Computer Engineering and the Ph.D. degree in Information Engineering from the University of Padova, Padua, Italy, in 2004 and 2007, respectively. From 2008 to 2010, he was a Postdoctoral Fellow with the Department of Mechanical Engineering, University of California at Santa Barbara, Santa Barbara, CA, USA. He is currently an Associate Professor with the Department of Information Engineering, University of Padova. His research interests include distributed algorithms for optimization, estimation and control over networks, nonparametric estimation and learning for robotics.

Alberto Dalla Libera received a Laurea degree in Control Engineering and the Ph.D. degree in information engineering from the University of Padua, Padua, Italy, in 2015 and 2019, respectively. He is currently a research fellow at the Department of Information Engineering of the University of Padua. His research interests include Robotics, Reinforcement Learning, Machine Learning, and Identification. In particular, he is interested in the application of Machine Learning techniques for modeling physical systems.

Diego Romeres (Senior Member, IEEE) received his M.Sc. degree (summa cum laude) in control engineering and the Ph.D. degree in information engineering from the University of Padua, Padua, Italy, in 2012 and 2017, respectively. He is currently a Principal Research Scientist and Team Leader of the Intelligent Robotics Team at Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. He held visiting research positions at TU Darmstadt, Darmstadt, Germany, and at ETH, Zurich, Switzerland. His research interests include robotics, artificial intelligence, machine learning, reinforcement learning, bayesian optimization and system identification theory.

Gianluigi Pillonetto was born on January 21, 1975 in Montebelluna (TV), Italy. He received the Doctoral degree in Computer Science Engineering cum laude from the University of Padova in 1998 and the PhD degree in Bioengineering from the Polytechnic of Milan in 2002. He is currently a Full Professor of Control and Dynamic Systems at the Department of Information Engineering, University of Padova. His research interests are in the field of system identification, estimation and machine learning. He has published around 90 papers on these research subjects in peer reviewed international journals and three books. From 2014 to 2016 he has been Associate Editor of Systems & Control Letters and IEEE Transactions on Automatic Control. He currently serves as Associate Editor for Automatica. In 2003 he received the Paolo Durst award for the best Italian Ph.D. thesis in Bioengineering, he was the 2017 recipient of the Automatica Prize, assigned every three years for outstanding contributions to control theory by the International Federation of Automatic Control (IFAC) and Automatica (Elsevier), and he was Plenary Speaker at System Identification IFAC Symposium in 2018. He has been elevated to IEEE Fellow in 2020 for contributions to System Identification.