



# Full Bayesian identification of linear dynamic systems using stable kernels

G. Pillonetto<sup>a</sup> and L. Ljung<sup>b,1</sup>

Edited by Yuedong Wang, University of California, Santa Barbara, CA; received November 15, 2022; accepted March 27, 2023 by Editorial Board Member H. Vincent Poor

System identification learns mathematical models of dynamic systems starting from input–output data. Despite its long history, such research area is still extremely active. New challenges are posed by identification of complex physical processes given by the interconnection of dynamic systems. Examples arise in biology and industry, e.g., in the study of brain dynamics or sensor networks. In the last years, regularized kernel-based identification, with inspiration from machine learning, has emerged as an interesting alternative to the classical approach commonly adopted in the literature. In the linear setting, it uses the class of stable kernels to include fundamental features of physical dynamical systems, e.g., smooth exponential decay of impulse responses. Such class includes also unknown *continuous* parameters, called hyperparameters, which play a similar role as the model *discrete* order in controlling complexity. In this paper, we develop a linear system identification procedure by casting stable kernels in a full Bayesian framework. Our models incorporate hyperparameters uncertainty and consist of a mixture of dynamic systems over a continuum spectrum of dimensions. They are obtained by overcoming drawbacks related to classical Markov chain Monte Carlo schemes that, when applied to stable kernels, are proved to become nearly reducible (i.e., unable to reconstruct posteriors of interest in reasonable time). Numerical experiments show that full Bayes frequently outperforms the state-of-the-art results on typical benchmark problems. Two real applications related to brain dynamics (neural activity) and sensor networks are also included.

system identification | regularization | Bayesian methods | sensor and brain networks

System identification builds mathematical models of dynamic systems starting from input–output measurements (1, 2). It has been around for more than half a century, with the term coined by Zadeh (3). Over the years, several approaches and frameworks for System Identification have been suggested. In the 1960s, two major avenues were laid out:

- Based on traditional statistics, it was suggested to parameterize the system model with finite-dimensional parameter structures and apply statistical techniques, like maximum likelihood or prediction error methods (PEM), to estimate the parameters. This can be called “the classical system identification framework” and is described in refs. 1 and 2. A recent extension to regularized/kernel criteria is treated in ref. 4.
- Realization-based techniques (“subspace methods”). States in state-space models are estimated directly using input–output measurements. These states are then used to construct state-space models with linear algebraic techniques. This approach is treated in, e.g., ref. 5.

Despite such long history, research in data-based dynamical modeling is still extremely active. Many complex physical processes, arising, e.g., in biology and industry, require an increasingly deep knowledge for improved prediction and control. This often requires to handle high-dimensional data, posing new challenges. Examples include the identification of complex physical systems like networks consisting of many interconnected dynamic systems, (6–9) for applications in engineering, biomedicine, and neuroscience.

In recent years, the regularization techniques described in refs. 4 and 10 have proved to be a powerful alternative to classical identification procedures based on PEM. Instead of postulating parametric structures, using the concept of discrete model order to control their complexity, the unknown dynamic system is directly searched for in a high-dimensional space. Ill-posedness is then circumvented by including some information on the physics of the problem. Resulting estimators look for solutions that balance adherence to experimental data and a penalty term accounting for dynamic systems

## Significance

System identification learns models of dynamical systems from input–output measurements. Estimated models should generalize by predicting system’s output responses to new, previously unseen inputs. The classical approach postulates different finite-dimensional structures, using the concept of discrete model order to control their complexity. Inspired by an alternative recent approach, called kernel-based identification, a Bayesian procedure for linear system identification is proposed. It reconstructs impulse responses posterior through careful use of marginalization. This permits to sample efficiently key hyperparameters accounting for systems stability. Obtained models are a mixture of systems over a continuum spectrum of dimensions. This can improve the generalization capability: State-of-the-art results are frequently outperformed on typical benchmark problems. Applications related to brain and sensor networks are also illustrated.

The authors declare no competing interest.

This article is a PNAS Direct Submission. Y. W. is a guest editor invited by the Editorial Board.

Copyright © 2023 the Author(s). Published by PNAS. This article is distributed under [Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 \(CC BY-NC-ND\)](https://creativecommons.org/licenses/by-nc-nd/4.0/).

<sup>1</sup>To whom correspondence may be addressed. Email: [lennart.ljung@liu.se](mailto:lennart.ljung@liu.se).

This article contains supporting information online at <http://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2218197120/-DCSupplemental>.

Published April 24, 2023.

features. Useful regularizers introduced in recent years embed notions of stability, e.g., smooth exponential decay of impulse responses whose convolutions with the inputs define the outputs in the linear and time-invariant setting (11–13). This makes the search space manageable by inducing a ranking of possible solutions: among dynamic systems that describe the data in a similar way, the one that is, in some sense, more stable will be selected.

Kernel methods are an important tool to induce the desired ranking of solutions (14), with roots in functional analysis (15, 16). A positive definite kernel induces a reproducing kernel Hilbert space (RKHS) whose functions inherit its properties. For instance, the so-called stable kernels are important in system identification since they induce RKHSs containing only absolutely summable impulse responses (17–19). There is also a fundamental relationship between regularization in RKHSs and Bayesian regression (20, 21) that arises when the kernel is interpreted as a covariance function (22–24). In the stochastic framework, the ranking of different solutions is induced by a Gaussian prior over a function space. We will follow this Bayesian viewpoint, focusing on linear system identification and adopting high-order autoregressive with exogenous inputs (ARX) models as search spaces. This allows us to formulate the kernel just in terms of a covariance matrix, modeling impulse responses as (zero-mean) Gaussian vectors.

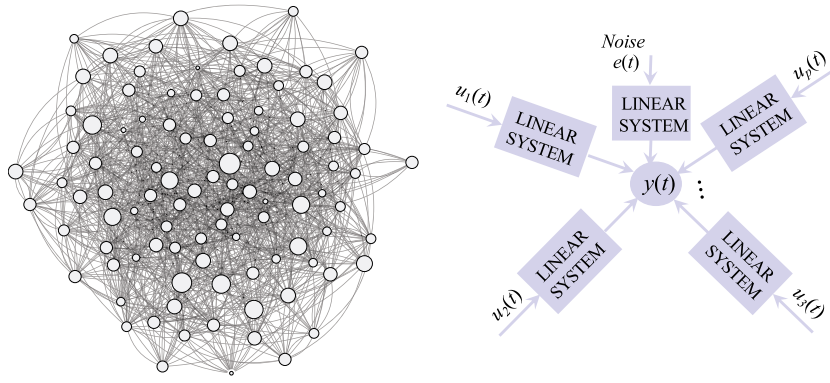
Our study focuses on the so-called stable spline/TC kernels introduced in refs. 11, 25 and 13. Such impulse response model was also derived through maximum entropy concepts in ref. 26 using the approach proposed by Jaynes to derive complete statistical prior distributions from incomplete a priori information (27). Among all the probabilistic descriptions that comply with some constraints, the maximum entropy criterion selects the one with the largest entropy. Such solution corresponds to the distribution that can be realized in the greatest number of ways according to Jaynes' concentration theorem. When exponential stability is the only available knowledge about a linear dynamic system, the maximum entropy prior is defined by the stable spline/TC kernel. It models impulse responses by means of Gaussian priors that include information on smooth exponential decay. Related covariances depend on positive scale factors  $\lambda_i$ , one assigned to each impulse response as a rule, and a (possibly) common decay rate  $\alpha$  assuming values over the unit interval. Such hyperparameters are typically unknown and have to be estimated from data. Their (continuous) tuning regulates model complexity, replacing the concept of (discrete) order selection encountered in the classical setting. For example, large  $\lambda_i$  associated with a value of  $\alpha$  close to one lead to complex stochastic models: impulse responses coefficients have large variance and decay to zero very slowly.

In Bayesian regularization, Empirical Bayes (EB) is one of the most used criteria to tune hyperparameters (28–31). It relies on the concept of marginal likelihood (ML) which, in our setting, corresponds to the total probability where all the impulse responses are integrated out. One important ML feature is its connection with the concept of equivalent degrees of freedom (32) which allows EB to incorporate Occam's factors (33, section 2). Hence, unnecessarily complex models can be automatically penalized. When identification data become available, ML is optimized w.r.t. the  $\lambda_i$  and  $\alpha$ . Next, hyperparameters are set to their estimates and impulse responses estimates become available in closed form. This defines the stable spline/TC estimator which has been proved to challenge classical PEM (10). It is an important option for linear system identification available in

the popular MATLAB system identification toolbox (34). A more sophisticated technique than EB studied in this paper interprets not only the impulse response but also the hyperparameters as random variables by introducing hyperpriors. This leads to an approach known as full Bayes (FB) in the literature and calls for calculation of minimum variance estimates which include hyperparameters uncertainty (35). EB finds justifications also in this setting: If hyperpriors are poorly informative, with the ML unimodal and well concentrated around its peak, EB will return impulse responses estimates close to the optimal ones [33, section 4]. However, while ML asymptotics have been well studied in the literature, e.g., refs. (31, 36–38) and 39 for studies on tail decay rates in the context of Gaussian processes, small samples properties are much harder to be understood (40, 41). Due to its nonconvex nature, the ML shape could be complex (possibly multimodal) also when using ARX models with a small number of inputs, a problem that can be further exacerbated in the study of dynamic networks. It is thus of interest to investigate whether FB may have advantages over EB in linear system identification. It is the purpose of this paper to show that this is the case.

Markov chain Monte Carlo (MCMC) is a key class of algorithms to implement FB. One of the most ubiquitous versions exploits the Metropolis–Hastings algorithm, e.g., refs. 35 and 42–44 for applications in system identification. MCMC algorithms first reconstruct the posterior of interest in sampled form by generating a suitable Markov chain. Then, minimum variance estimates are extracted by Monte Carlo integration. Since ARX models lead to regularized linear regression, the use of MCMC could appear straightforward in our context. Poorly informative distributions can be assigned to the hyperparameters and, following, e.g., ref. 45, an MCMC scheme similar to Gibbs sampling could be implemented. Even if theoretically correct, we will show that such approach does not work in practice due to some stable kernels peculiarities. In fact, as illustrated in [SI Appendix](#), impulse response realizations generated during an MCMC run can carry a huge amount of information on  $\alpha$ , hence inducing full-conditional distributions highly concentrated around their peak. This in practice undermines chain's irreducibility, i.e., the capability of visiting all the relevant parts of the posterior. Slow mixing affects the simulation, making it impossible to achieve convergence in a reasonable time. To solve this problem, we design a scheme based on the ML. Marginalization as a tool to enhance the effectiveness of simulation schemes has been pointed out also recently, e.g., to sample covariance matrices and in the presence of correlated latent variables (46, 47). Here, we show that interpreting impulse responses as nuisance parameters and integrating out from the joint posterior is crucial to sample efficiently  $\alpha$ . In addition, we prove that all the scale factors  $\lambda_i$  can still be updated in an automatic way, as happens when no marginalization is performed and Gibbs sampling is adopted. The proposed scheme does not fall exactly inside the classical MCMC class since some algorithmic parts do not use the Metropolis–Hastings algorithm. However, it permits to sample all the hyperparameters still guaranteeing convergence to the desired posterior.

The procedure is tested on artificial and real data. First, we use known benchmark problems taken from the system identification literature and show that FB leads to state-of-the-art generalization performance. It improves prediction capability over EB and also over the classical approach equipped with a special oracle-based procedure to tune model order. Advantages become more and more evident as system complexity (measured,



**Fig. 1.** Modern distributed sensors/agents are equipped with large storing capabilities and make available a large amount of data. Measurements are often generated by interconnected phenomena, as depicted in the left panel. Such complex physical systems arise in several fields of science and engineering. They can be artificial, like power grids and sensor/robots networks which cooperate to obtain a common goal, or can be encountered in biology, e.g., the brain network in neuroscience (6–9). They can be seen as a set of nodes associated with measurable (noisy) outputs  $y$ . As depicted in the right panel, each node can communicate with other nodes. Links can be defined, e.g., by time-invariant linear systems each characterized by a function known as impulse response in the literature. Such systems are driven by known inputs or nonmeasurable noises whose convolutions with the impulse responses define the output. Identification of these networked phenomena may unveil key information about their working and control. The algorithm proposed in this paper can be applied to estimate the linear dynamics which govern any node (line-by-line identification) when first-principle models are not conceivable or too difficult to obtain.

e.g., by the number of inputs) increases. This makes the proposed procedure appealing also for the identification of complex linear systems when first-principle models are not conceivable or too expensive/difficult to obtain (48–51) (Fig. 1 and related caption). As a proof of concept, two real applications are also considered: prediction of brain dynamics (neural activity) from blood oxygen-level-dependent time series and thermodynamic monitoring of a building through a wireless sensor network.

Readers who are not so familiar with concepts encountered in system identification find also a brief tutorial in the first part of [SI Appendix](#).

## Classical System Identification

**Modeling of Dynamical Systems: Three Main Players.** Modeling of dynamical systems is characterized by three key components.

The first one is a **family of parameterized models**. Any model can be seen as a map from past observed input–output data available at time  $t$ , denoted by  $\mathcal{Z}^t$ , to a prediction of the next (scalar, for simplicity) output  $y(t+1) \in \mathbb{R}$ , denoted by  $\hat{y}(t+1|\mathcal{Z}^t)$ . When the parameter vector  $\theta$  varies over a set  $D_\theta$ , the models define a *model structure*.

The second component is a **parameter estimation method** which determines the parameters on the basis of the observed data. The archetypical approach is to determine  $\theta$  as the one that minimizes the error between the observed outputs and the ones predicted by the model. Different loss functions  $V_n$  can be chosen to measure such discrepancy. An important example is the quadratic loss which leads to the following prediction error method (PEM)

Fit to a dataset  $\mathcal{Z}$ :

$$V_n(\theta, \mathcal{Z}) = \frac{1}{n} \sum_{i=1}^n (y(t) - \hat{y}(t|\theta, \mathcal{Z}^{t-1}))^2 \quad [1a]$$

Parameter estimate for an estimation dataset  $\mathcal{Z}_e$ :

$$\hat{\theta}_n = \arg \min_{\theta \in D_\theta} V_n(\theta, \mathcal{Z}_e). \quad [1b]$$

The third phase is a **validation process** where the model is validated or falsified. Model quality can be measured in terms of capability to predict unseen data contained in a validation

set  $\mathcal{Z}_v$ . Different prediction horizons  $h$  are also used for this purpose: Outputs  $y_v(t)$  in  $\mathcal{Z}_v$  are compared with the  $h$ -step ahead predictions  $\hat{y}_v^h(t) := \hat{y}_v(t|\hat{\theta}_n, \mathcal{Z}_v^{t-h})$  computed by the estimated model using the validation outputs only up to instant  $t-h$ . An useful generalization measure is also the  *$h$ -step ahead percentage prediction fit*:

$$\mathcal{F}_h = 100 \left( 1 - \frac{\|y_v - \hat{y}_v^h\|}{\|y_v - \bar{y}_v\|} \right), \quad [2]$$

where  $\bar{y}_v$  is the average value of the validation outputs. When only the inputs contained in  $\mathcal{Z}_v$  can be exploited by the model to predict future values, a **simulated output** is obtained with the related fit denoted by  $\mathcal{F}_\infty$ . In practice, since the output data in  $\mathcal{Z}_v$  cannot be used, the model calculates the simulation by assuming the system initially at rest at  $t=0$  and then replacing any  $y_v(t)$  with its  $t$ -step ahead prediction.

**Parametric Linear Model Structures.** A general model structure for multiple-input single-output (MISO) time-invariant linear systems is given by the transfer functions  $G_k$  from inputs  $u_k$  to output and the transfer function  $H$  from a white noise source  $e$  to output additive disturbances. In discrete time, using one time unit as sampling interval and  $q$  to denote the shift operator  $qy(t) = y(t+1)$ , one has

$$y(t) = \sum_{k=1}^p G_k(q, \theta) u_k(t) + H(q, \theta) e(t) \quad [3a]$$

$$\mathcal{E}e^2(t) = \sigma^2; \quad \mathcal{E}e(t)e(k) = 0 \text{ if } k \neq t, \quad [3b]$$

where  $\mathcal{E}$  denotes mathematical expectation. The *system impulse responses* are then obtained by expanding the  $G_k(q, \theta)$  and  $H(q, \theta)$  in the inverse (backwards) shift operator:

$$G_k(q, \theta) = \sum_{j=1}^{\infty} g_k(j, \theta) q^{-j}, \quad k = 1, \dots, p \quad [4]$$

$$H(q, \theta) = h_0(\theta) + \sum_{j=1}^{\infty} h(j, \theta) q^{-j}, \quad [5]$$

where  $h_0(\theta) = 1$  is typically assumed. In the above equations, the index  $j$  assumes only nonnegative values due to system causality and, given  $\theta$ , the scalars  $g_k(j, \theta)$  and  $h(j, \theta)$  are the impulse response coefficients of the deterministic and stochastic system part, respectively. For expressions of the associated  $h$ -step ahead predictors  $\hat{y}_k^h(t)$  and related predictor impulse responses, e.g., ref. [1, section 3]. The issue is how to parameterize  $G_k$  and  $H$ .

**Popular Black-Box Linear Models.** Popular *black box* (no physical insight or interpretation) parameterizations are to let  $G_k$  and  $H$  be rational in the shift operator:

$$G_k(q, \theta) = \frac{B_k(q, \theta)}{F_k(q, \theta)}; \quad H(q, \theta) = \frac{C(q, \theta)}{D(q, \theta)}, \quad [6]$$

where all the  $B_k, F_k, C$  and  $D$  are polynomials of  $q^{-1}$  while the parameter vector  $\theta$  contains the (unknown) polynomial coefficients. In many real applications, the polynomials order (which determine the dimension of  $\theta$ ) are also unknown and needs to be determined using  $Z_e$ .

*ARMAX models* are obtained letting  $F_k = D$  and represent fundamental descriptions of many dynamic systems encountered in nature (1). Another very common case is  $F_k = D$  and  $C = 1$  which gives the *ARX model*. It is the key component of our Bayesian model described below.

## Full Bayesian Linear System Identification

**ARX Parametrization.** With  $F_k = D$  and  $C = 1$  in Eq. 3a, one obtains the ARX model  $D(q)y = \sum_{k=1}^p B_k(q)u_k + e$ , or

$$\begin{aligned} y(t) &= \sum_{k=1}^p B_k(q)u_k(t) + (1 - D(q))y(t) + e(t) \\ &= \sum_{k=1}^{p+1} \sum_{r=1}^m b_k(r)u_k(t-r) + e(t), \end{aligned} \quad [7]$$

where  $b_k(r)$  are the coefficients of the polynomials  $B_k(q)$  for the  $p$  inputs,  $-b_{p+1}(r)$  are those related to the  $D(q)$  polynomial, the  $(p+1)$ -th input  $u_{p+1}(t) := y(t)$  is the output, and  $m$  is the order of the polynomials. As  $m$  increases, such formulation can approximate with arbitrary accuracy any linear system in Eq. 3.

It is now useful to introduce the (column) vectors  $\theta_k$  which contain the polynomial coefficients  $\{b_k(r)\}_{r=1}^m$  and are associated with the  $p+1$  predictor impulse responses. We also build the vector  $Y$  with the observations  $y(t)$ , for  $t = 1, \dots, n$ , and  $p+1$  Toeplitz matrices  $\Phi_k \in \mathbb{R}^{n \times m}$  using the input-output data as follows:

$$\Phi_k = \begin{pmatrix} u_k(0) & u_k(-1) & \dots & u_k(-m+1) \\ u_k(1) & u_k(0) & \dots & u_k(-m+2) \\ \dots & \dots & \dots & \dots \\ u_k(n-1) & u_k(n-2) & \dots & u_k(n-m) \end{pmatrix}, \quad [8]$$

for  $k = 1, \dots, p+1$ . Note that the first  $p$  matrices contain the inputs while the last one is built using only the outputs since  $u_{p+1}(t)$  corresponds to  $y(t)$ . We can now rewrite Eq. 7 in matrix-vector form as follows

$$Y = \left( \sum_{k=1}^{p+1} \Phi_k \theta_k \right) + E = \Phi \theta + E, \quad [9]$$

where each  $\Phi_k \theta_k$  contains convolutions between the predictor impulse response  $\theta_k$  and past input or output data,  $\theta$  gathers all the impulse responses coefficients,  $\Phi$  is assumed full rank to simplify exposition, and the noise vector  $E$  is Gaussian.

**Full Bayesian Model.** ARX models are easy to estimate but may suffer from high variance. Bayesian regularization is adopted to face this difficulty by assigning to  $\theta$  a prior distribution. As anticipated in Introduction, we adopt a full Bayesian model where the hyperparameters which define such prior are also seen as random variables.

Conditional on the knowledge of their covariance matrices  $\Sigma_k$ , the  $\theta_k$  are zero-mean and independent Gaussian vectors, i.e.,

$$\theta_k | \Sigma_k \sim \mathcal{N}(0, \Sigma_k), \quad k = 1, \dots, p+1, \quad [10]$$

and, using  $\perp$  to indicate statistical independence,

$$\theta_i | \Sigma_i \perp \theta_j | \Sigma_j, \quad i \neq j.$$

Our covariances  $\Sigma_k$  are stochastic matrices that have to embed an important feature of stable physical systems: Predictor impulse responses are expected to decay smoothly and exponentially to zero. They depend on scale factors  $\{\lambda_k^2\}_{k=1}^{p+1}$  and a common decay parameter  $\alpha$  which form a set of mutually independent random variables. Each of the  $p+1$  objects  $\Sigma_k$  could be also assigned a different decay rate, making  $\alpha$  a vector of dimension  $p+1$ . This leads to minor modifications in the algorithm presented later on but may lead to improved models, as commented upon in “Material and Methods”.

Conditional on  $\alpha$  and its scale factor  $\lambda_k^2$ , the covariance  $\Sigma_k$  is perfectly known, being defined by the stable spline/TC kernel as follows:

$$\Sigma_k | \lambda_k^2, \alpha = \lambda_k^2 K_\alpha, \quad [11]$$

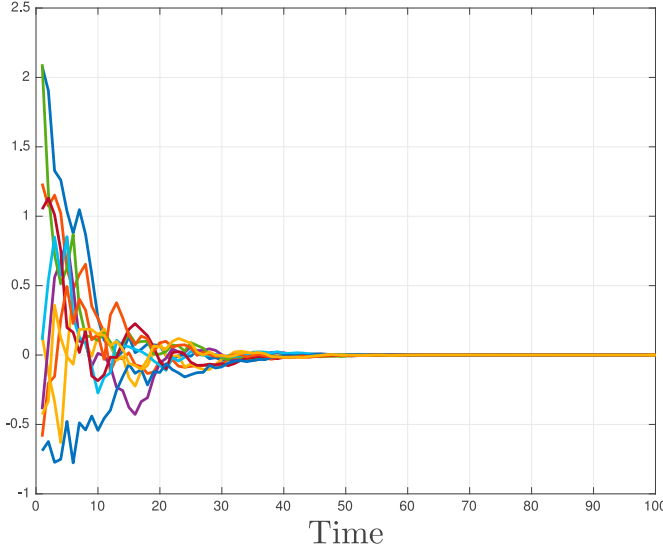
where  $K_\alpha$  is an  $m \times m$  matrix with  $i, j$  entry

$$K_\alpha(i, j) = \alpha^{\max(i, j)}. \quad [12]$$

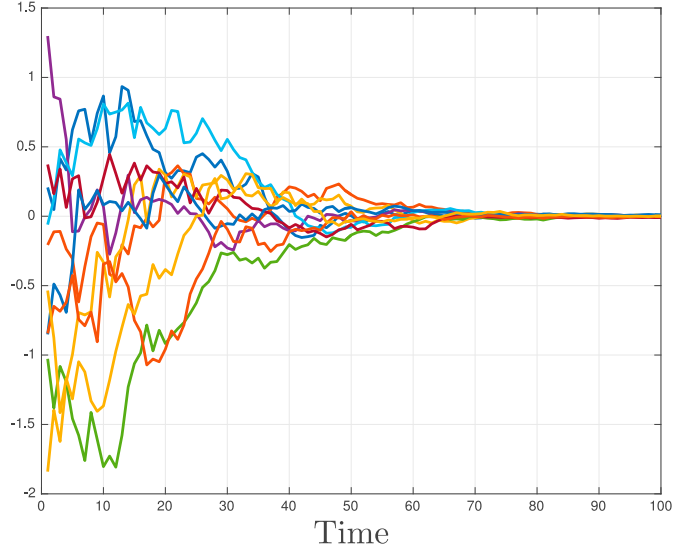
In this way,  $\alpha$  regulates how fast the impulse response variance goes to zero as time progresses. A simple way to understand the nature of our impulse response prior is to draw some realizations from a zero-mean Gaussian vector with stable spline covariance. This is done in Fig. 2 for two different decay rates  $\alpha$  and with the scale factor set to one. One can see that the stability parameter determines the effective dimension of the model. Hence, the choice of the size  $m$  of each  $\theta_k$  is not problematic as in the classical system identification framework. One has just to set it to a value as large as possible to capture predictor dynamics, compatibly with the computational resources. This choice can be also driven by the available information on the specific application under study. It may suggest which  $m$  can be seen as an upper bound on the number of past inputs and outputs able to influence significantly the next output value.

The predictor model is now fully specified by assigning hyperpriors not only to the scale factors  $\lambda_k^2$  and the decay rate  $\alpha$ , but even to  $\sigma^2$  in Eq. 3b. In fact, the noise variance affects the model and so it is natural to also treat it as a hyperparameter. All of these random variables are assumed mutually independent, i.e.,  $\sigma^2 \perp \alpha$  and  $\sigma^2 \perp \lambda_k^2$  for any  $k$ . The stability parameter  $\alpha$  is modeled as a uniform random variable on the unit interval. The scale factor  $\lambda_k^2$  is instead given the well-known Jeffrey’s prior (52), widely used to include (in practice) only nonnegativity information. It follows Jeffrey’s “noninformative prior finding

Impulse response realizations with  $\alpha = 0.8$



Impulse response realizations with  $\alpha = 0.9$



**Fig. 2.** Linear and time-invariant dynamic systems are characterized by their impulse responses. Given any input, such functions allow to compute the corresponding system output through convolutions. In this paper, impulse responses are seen as stochastic zero-mean Gaussian vectors with covariance proportional to the stable spline kernel in Eq. 12. The figure reports some realizations drawn from the stable spline prior when  $\alpha = 0.8$  (Left) and  $\alpha = 0.9$  (Right). One can see that the model encodes information on smooth exponential decay of impulse responses of stable dynamic systems. The parameter  $\alpha$  assumes values on the unit interval and regulates the decay rate. Values close to one indicate that the impulse response goes to zero slowly.

principle” related to invariance under monotone transformations, e.g., refs. 53 and 54 for other interesting discussions about this topic. One obtains that the (improper) probability density function  $p(\lambda_k^2)$  is proportional to  $1/\lambda_k^2$ , hence we can write

$$\alpha \sim \mathcal{U}(0, 1), \quad \lambda_k^2 \sim \frac{1}{\lambda_k^2}. \quad [13]$$

The measurement noise  $E$  is assumed independent of all the  $\theta_k$  and, conditional on the variance  $\sigma^2$ , one has

$$E|\sigma^2 \sim \mathcal{N}(0, \sigma^2 I_n), \quad [14]$$

where  $I_n$  is the  $n \times n$  identity matrix. The noise variance is also given a Jeffrey’s prior, i.e.,

$$\sigma^2 \sim \frac{1}{\sigma^2}. \quad [15]$$

Another advantage of Jeffrey’s priors, exploited later on, is that the posteriors of  $\sigma^2$  and  $\lambda_k^2$  conditional on  $\theta$  are easy to sample since they correspond to inverse-gamma distributions.

## Empirical and Full Bayes Estimators

**Structure of the Minimum Variance Estimate.** Let the hyperparameter vector  $\eta$  contain  $\{\lambda_j^2\}_{j=1}^{p+1}$ ,  $\alpha$  and  $\sigma^2$ . The a priori probability density function  $p(\theta, \eta)$  is thus defined by Eqs. 10, 13, and 15. We assume that it summarizes all the knowledge on impulse responses and hyperparameters before observing the first measurement contained in the output vector  $Y$ , [12, section 5.1].

Our target is to compute the posterior  $p(\theta|Y)$  and to extract from it the predictor impulse responses estimates

$$\mathcal{E}(\theta|Y) = \int \theta p(\theta|Y) d\theta. \quad [16]$$

However, this integral is analytically intractable also because marginalization w.r.t.  $\eta$  is first needed. This is highlighted by the following posterior reformulation

$$p(\theta|Y) = \int p(\theta, \eta|Y) d\eta = \int p(\theta|\eta, Y) p(\eta|Y) d\eta. \quad [17]$$

The integral on the r.h.s. of Eq. 17 contains two important factors: the marginal hyperparameters posterior  $p(\eta|Y)$  and the conditional distribution  $p(\theta|\eta, Y)$ . This latter term has a simple structure. Bayes rule shows that the probability density function of  $\theta$  conditional on  $\eta$  and  $Y$  is still Gaussian and available in closed-form, [12, Appendix]. In fact, letting  $\Sigma_\eta$  be the prior covariance of  $\theta$  conditioned on  $\eta$ , i.e.,

$$\Sigma_\eta = \text{blkdiag}(\lambda_1^2 K_\alpha, \dots, \lambda_{p+1}^2 K_\alpha), \quad [18]$$

one has

$$\mathcal{E}(\theta|\eta, Y) = \int \theta p(\theta|\eta, Y) d\theta \quad [19a]$$

$$= \left( \frac{\Phi^\top \Phi}{\sigma^2} + \Sigma_\eta^{-1} \right)^{-1} \frac{\Phi^\top}{\sigma^2} Y \quad [19b]$$

$$= \Sigma_\eta \Phi^\top \left( \Phi \Sigma_\eta \Phi^\top + \sigma^2 I_n \right)^{-1} Y. \quad [19c]$$

The last two expressions are equivalent but have different computational cost, with Eq. 19b to be preferred as the dataset size  $n$  grows. The formulation in Eq. 19c gives instead useful information on the structure of Eq. 19a. In fact, let  $v_\eta(i) \in \mathbb{R}^{(p+1)m}$  be the  $i$ -th column of  $\Sigma_\eta \Phi^\top$  and let  $c_\eta(i) \in \mathbb{R}$  be the  $i$ -th component of (the column vector)  $(\Phi \Sigma_\eta \Phi^\top + \sigma^2 I_n)^{-1} Y$ . Then, Eq. 19c can be rewritten as

$$\mathcal{E}(\theta|\eta, Y) = \sum_{i=1}^n c_\eta(i) v_\eta(i), \quad [20]$$

a result that can be also seen as the finite-dimensional version of the famous representer theorem (55). In view of the stable spline kernel structure reported in Eq. 12, the time-course of any basis vector  $v_\eta(i)$  is regulated by the stability parameter  $\alpha$ .

Let us now reconsider the minimum variance estimate in Eq. 16 that, differently from Eq. 20, incorporates also hyperparameters uncertainty. Combining Eqs. 17 and 20, one has

$$\mathcal{E}(\theta|Y) = \sum_{i=1}^n \int c_\eta(i) v_\eta(i) p(\eta|Y) d\eta. \quad [21]$$

By exploiting EB and combinations of covariances that include many different decay parameters, previous works on regularized system identification like (56, 57) showed improved performance over the stable spline/TC kernels. Eq. 21 points out that a new dimension is already reached by the Full Bayes estimate. In fact,  $\mathcal{E}(\theta|Y)$  is sum of  $n$  basis vectors that automatically embed a continuous spectrum of different  $\alpha$  values weighted by the marginal  $p(\eta|Y)$ . Now, the issue is to numerically compute such estimates.

**Empirical Bayes.** The marginal hyperparameters posterior present inside the integral in Eq. 17 satisfies

$$p(\eta|Y) \propto p(Y|\eta)p(\eta), \quad [22]$$

where  $p(Y|\eta)$  is the *marginal likelihood*, already indicated with ML. Using Laplace's integration, as described, e.g., in [12, Appendix], one proves that

$$-\log p(Y|\eta) = \frac{1}{2} Y^\top \Sigma_Y^{-1} Y + \frac{1}{2} \log \det 2\pi \Sigma_Y, \quad [23]$$

where

$$\Sigma_Y = \Phi \Sigma_\eta \Phi^\top + \sigma^2 I_n,$$

for large  $n$ , equivalent but more efficient expressions, along with their numerical implementations, are discussed in ref. 34. In Eq. 23, the last term given by  $\log \det 2\pi \Sigma_Y$  is the Occam's factor mentioned in Introduction which controls complexity. The ML hyperparameters estimate is then given by

$$\eta^{\text{ML}} = \arg \min_{\eta} \frac{1}{2} Y^\top \Sigma_Y^{-1} Y + \frac{1}{2} \log \det 2\pi \Sigma_Y. \quad [24]$$

This corresponds to using Eq. 1b, where  $\eta$  replaces  $\theta$  and the loss function  $V_n$  has become the minus log marginal likelihood. EB first solves Eq. 24 and then computes Eq. 19 setting  $\eta$  to its estimate  $\eta^{\text{ML}}$  as summarized in Algorithm 1.

---

#### Algorithm 1: Empirical Bayes (EB)

---

1. Compute the hyperparameters estimate  $\eta^{\text{ML}}$  by solving the optimization problem in Eq. 24;
  2. set  $\eta$  to  $\eta^{\text{ML}}$  and compute the impulse response estimates using Eq. 19b or, equivalently, Eq. 19c.
- 

If data  $Y$  are sufficiently informative, since the prior on  $\eta$  is rather flat, one can expect  $p(\eta|Y)$  in Eq. 22 to be quite concentrated around  $\eta^{\text{ML}}$ . This permits to interpret EB as a tool for approximating the true posterior in Eq. 17 via

$$p(\theta|Y) \simeq p(\theta|\eta^{\text{ML}}, Y).$$

This however points out that the multiresolution features enjoyed by Eq. 16, e.g., in terms of decay parameters as described after Eq. 21, can never be obtained by EB.

**Full Bayes.** It is useful to indicate with  $\mathcal{V}$  the random vector containing all the unknown variables  $\theta, \sigma^2, \alpha$  and  $\{\lambda_k^2\}_{k=1}^{p+1}$ . The notation  $\mathcal{I}_g(a, b)$  denotes the inverse-gamma of parameters  $a, b$  with probability density function

$$p(x) \propto x^{-a-1} e^{-b/x}, \quad x > 0. \quad [25]$$

Algorithm 2 describes our Full Bayes strategy. It builds a suitable Markov chain and returns an approximation  $\hat{\theta}$  of the minimum variance estimate reported in Eq. 16. One feature of our FB algorithm is the use of ML to sample  $\alpha$  in Step 3. Another one is that the samples  $\theta^{(i)}$  generated at any run and used to update the scale factors are not part of the chain's state (as in the classical MCMC schemes mentioned in Introduction). Indeed, they are not used in Step 3. However, such  $\theta^{(i)}$  reconstruct the posterior of  $\theta$  in sampled form, as reported in the following Proposition whose proof is discussed in SI Appendix.

**Proposition 1.** Let  $\{\theta^{(i)}\}_{i=1}^M$  and  $\{\mu^{(i)}\}_{i=1}^M$  be the vectors returned by Algorithm 2. Let  $\chi_A(\cdot)$  be the indicator function of a generic set  $A \subset \mathbb{R}^{m(p+1)}$  and let  $\mathbb{P}$  denote probability. Then, for all starting points of the chain (excluded a set of null measure w.r.t. the posterior of impulse responses and hyperparameters) and any set  $A$ , the following convergences hold with probability one:

$$\begin{aligned} \lim_{M \rightarrow \infty} \sum_{i=1}^M \frac{\chi_A(\theta^{(i)})}{M} &\rightarrow \mathbb{P}(\theta \in A|Y), \\ \lim_{M \rightarrow \infty} \sum_{i=1}^M \frac{\mu^{(i)}}{M} &\rightarrow \mathcal{E}(\theta|Y). \end{aligned} \quad [26]$$

SI Appendix also illustrates how Algorithm 2 is (in some sense) necessary to implement FB in stable kernel-based linear system identification since it overcomes critical flaws of classical MCMC schemes.

## Numerical Experiments

**Three Estimators at Stake.** We report results coming from simulated and real experiments. In any experiment, data are divided into an estimation ( $\mathcal{Z}_e$ ) and a validation ( $\mathcal{Z}_v$ ) dataset. The latter has to be interpreted as a container of future data that could not be used to estimate the model. Then, the performance of an identification procedure is measured by the  $h$ -step ahead percentage prediction fits  $\mathcal{F}_h$  defined in Eq. 2.

Three estimators will be adopted. The first one, denoted by **PEM-Or**, provides a useful reference on prediction performance. It relies on the classical system identification framework coupled with an oracle for model order selection. In statistical literature, the term oracle often indicates information about model properties coming from ideal/unrealistic sources. For PEM-Or, the validation set and the related fits  $\mathcal{F}_k$  are an "approximate oracle". They permit to control the model discrete dimension by maximizing a proxy for the average prediction capability of future data. Specifically, PEM-Or uses ARMAX structures of different discrete dimensions  $d$  with polynomial orders ranging from 1 to 30. Any structure is fitted to the estimation data  $\mathcal{Z}_e$  and the prediction fits  $\mathcal{F}_h$  (which turn out function of  $d$ ) are

## Algorithm 2: Full Bayes (FB)

**Initialization:** Let  $\theta^{(0)}$  be the initial impulse responses values, e.g., the least squares estimate  $\theta^{(0)} = (\Phi^\top \Phi)^{-1} \Phi^\top Y$ . Let also  $\alpha^{(0)}$  be the initial value of the decay rate  $\alpha$ , e.g.,  $\alpha^{(0)} = 0.9$ , and let  $\Delta$  be the SD of the random walk used in Step 3, e.g.,  $\Delta = 0.02$ .

For  $i = 1, 2, \dots, M$ , repeat the following steps.

1. Update the noise variance by setting  $\sigma^{2(i)}$  to a sample drawn from the conditional density

$$\sigma^2 | (\mathcal{V} \setminus \sigma^2, Y) \sim \mathcal{I}_g\left(\frac{n}{2}, \frac{1}{2} \|Y - \Phi \theta^{(i-1)}\|^2\right).$$

where  $\mathcal{I}_g(a, b)$  is the inverse-gamma of parameters  $a$  and  $b$  with pdf given in Eq. 25.

2. Build  $K_\alpha$  through Eq. 12 with  $\alpha = \alpha^{(i-1)}$ . Update the scale factors by setting each  $\lambda_k^{2(i)}$  to a sample drawn from the conditional density

$$\lambda_k^2 | (\mathcal{V} \setminus \lambda_k, Y) \sim \mathcal{I}_g\left(\frac{m}{2}, \frac{(\theta_k^{(i-1)})^\top K_\alpha^{-1} \theta_k^{(i-1)}}{2}\right).$$

3. Let  $a = \alpha^{(i-1)} + \Delta z^{(i)}$ , where  $z^{(i)}$  is zero-mean Gaussian with unit variance (all the  $\{z^{(i)}\}_{i=1}^M$  are mutually independent). If  $a < 0$  or  $a \geq 1$ , set the acceptance probability  $p = 0$ , otherwise

$$p = \min\left(1, \frac{p(Y|\eta_a)}{p(Y|\eta_b)}\right),$$

where  $\eta_a, \eta_b$  are the hyperparameter vectors containing  $\sigma^{2(i)}$ , the scale factors  $\lambda_k^{2(i)}$  and, respectively, the decay rate  $a$  and  $\alpha^{(i-1)}$  while the marginal likelihood  $p(Y|\eta)$  is defined by Eq. 23 for any  $\eta$ . Finally, set  $\alpha^{(i)} = a$  with probability  $p$ , otherwise, let  $\alpha^{(i)} = \alpha^{(i-1)}$ .

4. Build  $\Sigma_\eta$  using Eq. 18 with the hyperparameter vector  $\eta$  containing  $\sigma^{2(i)}$ , the scale factors  $\lambda_k^{2(i)}$  and  $\alpha^{(i)}$ . Then, define

$$\mu^{(i)} = \left(\frac{\Phi^\top \Phi}{\sigma^2} + \Sigma_\eta^{-1}\right)^{-1} \frac{\Phi^\top Y}{\sigma^2},$$

and let  $\theta^{(i)}$  be a sample drawn from the conditional density

$$\theta | (\mathcal{V} \setminus \theta, Y) \sim \mathcal{N}(\mu^{(i)}, \hat{\Sigma}), \quad \text{where} \quad \hat{\Sigma} = \left(\frac{\Phi^\top \Phi}{\sigma^2} + \Sigma_\eta^{-1}\right)^{-1}.$$

Return the  $\{\theta^{(i)}\}_{i=1}^M$  as the sampled version of  $p(\theta|Y)$  and the impulse response estimates as  $\hat{\theta} = \frac{1}{M} \sum_{i=1}^M \mu^{(i)}$ .

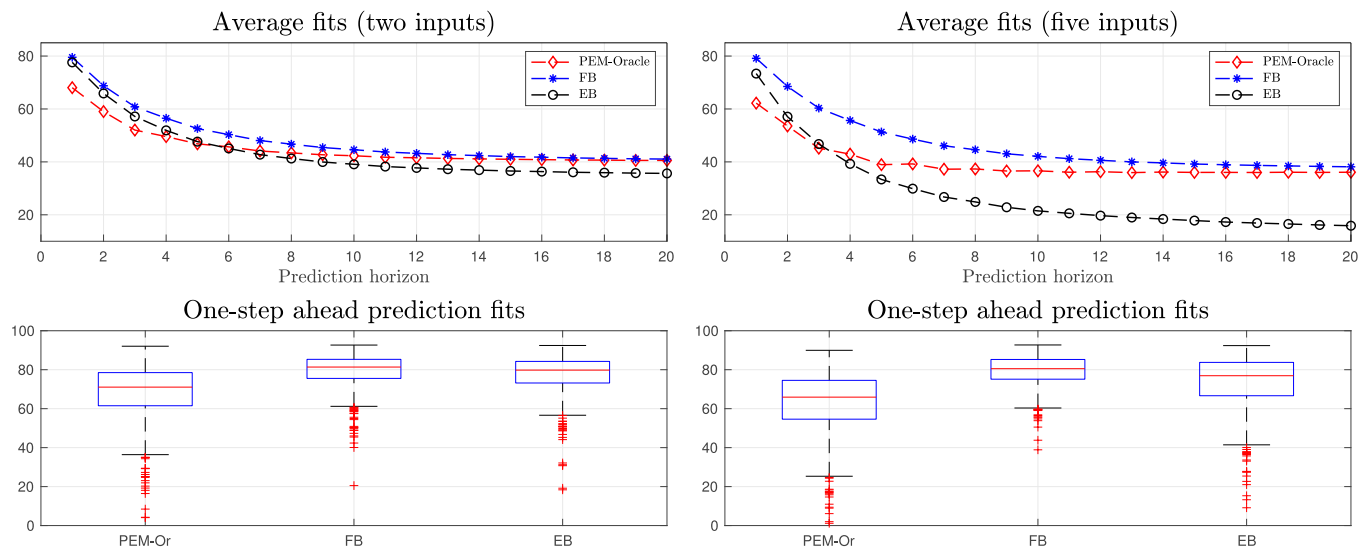
computed for  $h = 1, \dots, 20$ . Next, the procedure has access to  $\mathcal{Z}_v$  to control complexity: The dimension  $d$  is selected as the one maximizing the average value of the prediction fits. Hence, in light of the nature assigned to  $\mathcal{Z}_v$ , PEM-Or has to be seen as an ideal scheme not implementable in real applications.

The second estimator **EB** uses the Empirical Bayes scheme summarized in Algorithm 1. Finally, the third estimator is denoted by **FB** and implements the new Full Bayes procedure defined by Algorithm 2. Note that, differently from **PEM-Or**, **EB** and **FB** are implementable in practice since they use only the data contained in  $\mathcal{Z}_e$  to estimate the model. Some implementation details regarding the three identification procedures can be found in *Material and Methods*.

**Benchmark Problems.** Simulated data are used to test the three estimators through some known benchmark problems proposed in ref. 10. Two Monte Carlo studies of 1,000 runs are considered. At any run, an ARMAX model of order 30 is randomly generated (details are in *Material and Methods*). In the first and second study, the number of inputs  $p$  in Eq. 3 is set to 2 and 5, respectively. System input is different at any run and is realization of white Gaussian noise of unit variance. The datasets  $\mathcal{Z}_e$  and  $\mathcal{Z}_v$  contain, respectively, 300 and 1,000 input–output pairs collected after getting rid of initial conditions effect. The rationale underlying the large size of the validation set is to obtain a good proxy for the average prediction performance of the estimated models on unseen data. When using EB and FB, the dimension of each  $\theta_k$  is set to 40. Increasing this dimension does not influence obtained results, as detailed in the last part of *SI Appendix*. To deal with initial conditions effects, the first 40 input–output pairs in  $\mathcal{Z}_e$  are used just as entries of the regression matrix.

Fig. 3 displays the mean of the fits  $\mathcal{F}_h$  as a function of the prediction horizon  $h$  (*Top* panels) and the MATLAB boxplots of the 1,000 values of  $\mathcal{F}_1$  (*Bottom* panels). It is apparent that the prediction capability of FB is superior than that of EB. Advantages become more evident when the number of inputs augments (*Right* panels). This can be explained considering that FB is theoretically immune to the presence of local minima when tuning hyperparameters and system estimates can account for all the ML complexity. Such features can become more and more important as the problem dimension increases. Interestingly, FB outperforms also PEM-Or. This is remarkable since the oracle is an ideal tuning which has access to  $\mathcal{Z}_v$  to control complexity. The explanation here is that PEM-Or selects the model order balancing bias and variance among a finite set of given models. FB deals with such trade-off using a continuous set of regularization parameters. In addition, it provides a system estimate which takes into account all of their uncertainty, averaging over a continuous spectrum of dimensions. In this way, it can obtain better-performing trade-offs. Furthermore, one has to consider that these benchmark problems use a dataset  $\mathcal{Z}_e$  of relatively small size (300) and quite complex systems (ARMAX of orders 30), a situation where advantages of regularization emerge more clearly.

**Brain Dynamics: Neural Activity Prediction.** Blood oxygen level–dependent (BOLD) time series measured in a brain region (often through fMRI) are strongly connected with neural activity (58) (Fig. 4). Prediction of BOLD signals, obtained also exploiting data collected in adjacent regions, is important, e.g., for control and therapeutic purposes (59, 60). Obtained models give insights into brain connectivity, with lack of links among regions which can detect the onset (or consequences) of a disease (9, 61, 62). The Dynamic Causal Model (DCM), originally developed in ref. 63 to describe such relationships, is formulated in state space whose dimension equals the number of considered brain areas. Any state component describes the neural activity in a region. The inputs may be experimentally designed (task-related stimuli)



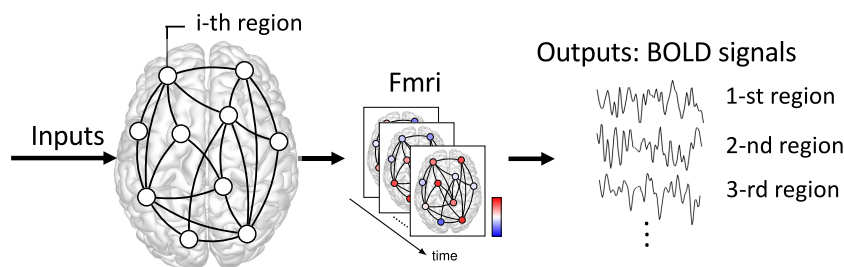
**Fig. 3.** Identification of simulated ARMAX models. *Top* Average of the  $h$ -step ahead fits  $\mathcal{F}_h$  as defined in Eq. 2. *Bottom* Boxplots of the 1,000 values of  $\mathcal{F}_1$ . Recall that PEM+Or uses additional information, having access to the validation set to control model complexity.

or just stochastic noises describing random neural fluctuations (resting-state) (64). Finally, the output in a region is the BOLD signal modeled as a nonlinear transformation of its neural activity corrupted by additive noise (58), expressed as fraction of signal change w.r.t. the basal value. We consider the same dataset described in (9, section 2.4) consisting of BOLD signals associated with seven brain regions simulated via DCM. Our problem is to predict the output of the first region considering the time series coming from the surrounding areas as six inputs to an ARMAX or regularized ARX model. This linear setting finds support also in real applications where linear models are known to give good approximations of the actual neural processes even if brain dynamics are known to be nonlinear (59).

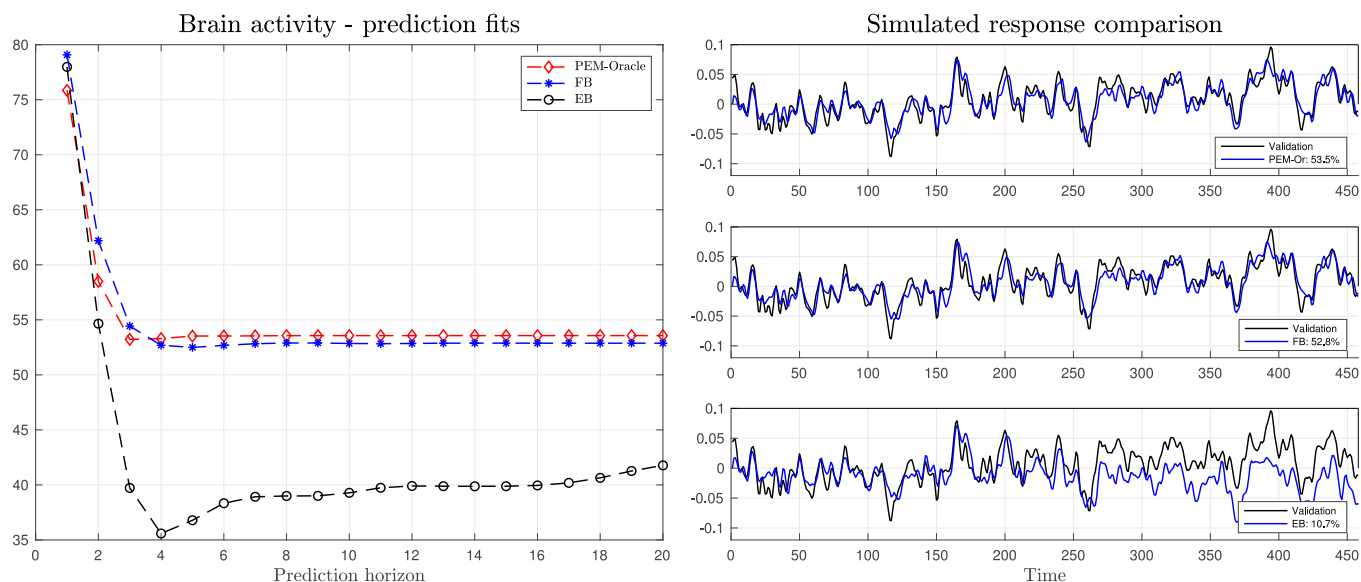
The dataset contains around 900 samples for any of the 7 time series collected in the cerebral regions with sampling time 1.5 s. It is split into an identification set  $\mathcal{Z}_e$  and a validation set  $\mathcal{Z}_v$  of equal size. The three identification procedures are implemented with the same settings used to solve the benchmark problems. In this study, the choice  $m = 40$  corresponds to a temporal interval of 1 min. We have also repeated the experiment with  $m = 100$ , a large value to retain the relevant temporal dependencies (65), obtaining the same results described in the next lines. The prediction fits of brain activity are in *Left* panel of Fig. 5. FB generalization capability is close to that of the oracle and outperforms EB. The simulated outputs are shown in *Right* panels: the percentage simulation fits  $\mathcal{F}_\infty$  of FB and PEM-Or are close to 53% while those of EB are only around 10%. The reason is that the dynamic model estimated by EB has a component close to instability (a

root of the polynomial  $D$  in Eq. 6 is much close to 1), and this deteriorates the generalization performance for large prediction horizons.

**Sensor Networks: Temperature Prediction.** The algorithms are now tested on real data regarding thermodynamic modeling of buildings. A wireless sensor network of 24 *Tmote-Sky* nodes produced by Moteiv Inc was placed in a small residential building of about 300 m<sup>2</sup>. The first node collects temperature measurements (seen as the output of a linear system) and has to predict future values on the basis of the profiles of either relative humidity, or temperature, measured in Celsius, sent by the other sensors (seen as the system inputs). This problem is relevant, e.g., for model predictive control and optimization of energy use (66–68). The measurement period lasted for 8 d with the temperature values ranging between 5 °C and 30 °C (sensors were also placed outdoor and near a radiator). The building was inhabited with the heating system controlled by a thermostat manually set every day depending upon occupancy and other needs. The monitoring period is rather small and does not permit to obtain a model which describes seasonal variations. Hence, a “stationary” environment is assumed and ARMAX or regularized ARX models with 1 output and 23 inputs are used. The sampling time is 8 min, and data consist of 1,400 samples for the output and any input. First, they are normalized, so as to have zero mean and unit variance. Then, they are divided into an identification set  $\mathcal{Z}_e$  and a validation set  $\mathcal{Z}_v$  of equal size.



**Fig. 4.** Blood oxygen level-dependent (BOLD) time series measured via fMRI in different brain regions are connected with neural activity. This latter can be induced, e.g., by external stimuli related to a task to perform and/or random neural fluctuations.



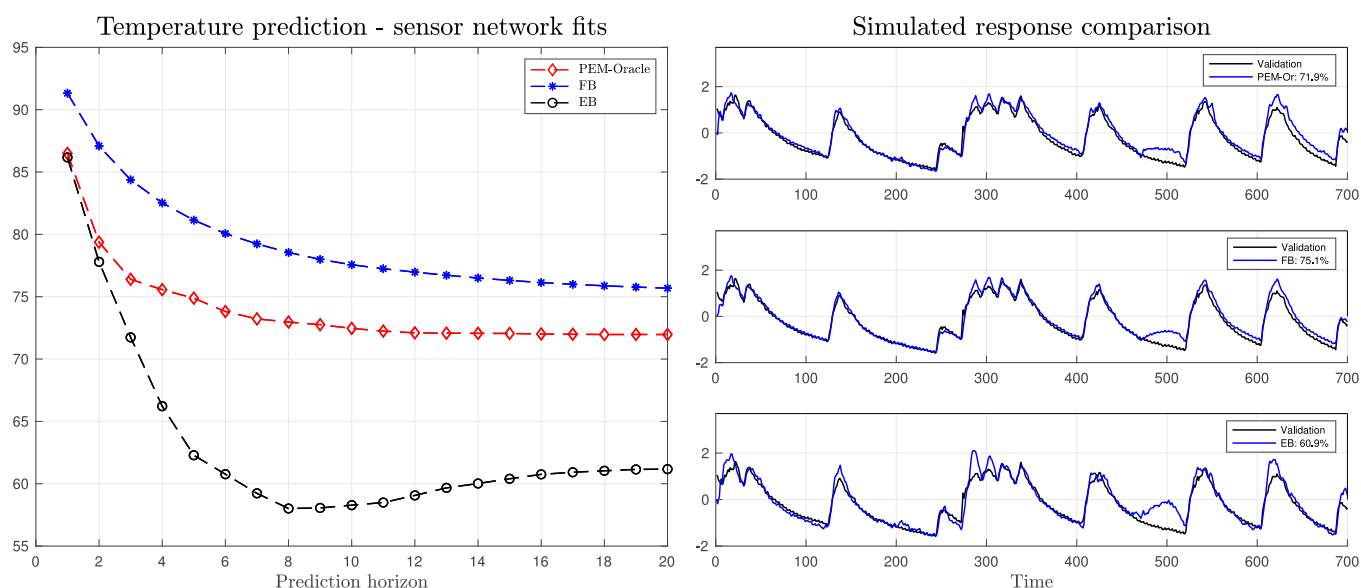
**Fig. 5.** Brain activity prediction. *Left* Fits  $\mathcal{F}_h$  as defined in Eq. 2. *Right* Simulated outputs by PEM-Or (Top), FB (Middle), and EB (Bottom).

The temperature prediction fits of the sensor network are reported in *Left* panel of Fig. 6 using the three identification procedures with the same settings previously described. Note that the predictor length  $m = 40$  corresponds to a large temporal window of more than 5 h. Results are in line with those obtained in the benchmark problems: FB outperforms EB and PEM-Or. The simulated outputs are displayed in the right panels: the percentage simulation fit  $\mathcal{F}_\infty$  of FB is close to 75% while those of EB and PEM-Or are, respectively, around 61% and 72%.

## Discussion

Control of complexity is crucial in system identification, with profound implications for model prediction capability. In this paper, we have shown that model selection for Bayesian linear system identification, which involves continuous tuning of hyperparameters, finds an interesting dimension by replacing

EB with FB. This latter is implemented through a stochastic simulation scheme. It faces some critical issues related to the use of classical MCMC schemes coupled with impulse responses priors which embed exponential stability. The identification procedure here described can now account for all the ML shape, overcoming nonconvexity issues and making complexity control even more robust. This reflects also on the structure of impulse responses estimates which can now incorporate a continuum spectrum of different stability parameters. This increases the expressive power of the regularized estimator, improving its ability to simulate and predict complex linear dynamic systems. We also envision extensions of the identification procedure here proposed to the nonlinear setting (69). This could be, e.g., obtained by replacing the stable kernels here treated with nonlinear versions (24), like the ones described in refs. 70 and 71 which embed some fading memory properties present in many real dynamic systems.



**Fig. 6.** Temperature prediction using a real sensor network. *Left* Fits  $\mathcal{F}_h$  as defined in Eq. 2. *Right* Simulated outputs by PEM-Or (Top), FB (Middle), and EB (Bottom).

## Materials and Methods

Given an estimated model and a validation dataset  $\mathcal{Z}_v$ , the percentage prediction fits  $\mathcal{F}_h$  in Eq. 2 are computed by the function **compare** of the MATLAB system identification toolbox with the system at rest (null initial conditions). This routine allows to compute the prediction over any horizon  $h$ , with the simulated output obtained by setting  $h$  to infinity.

PEM-Or is implemented through the command **armax**. The procedure EB estimates the regularized ARX model using **arxRegul** and **arx**. These routines tune the hyperparameters by estimating one different decay parameter  $\alpha$  for each predictor impulse response. This can be easily obtained also through FB with a simple modification of Step 3 of Algorithm 2: it is sufficient to update separately a different  $\alpha$  for each covariance of  $\theta_k$ . This more sophisticated version of FB has been also implemented, assessing that all the results remain essentially the same. For instance, in the benchmark problems, the average fits increase by around 1% using multiple decay rates.

In any experiment, the number of posterior samples generated by FB is 10,000. Such chain length has always ensured that quantiles {0.025, 0.25, 0.5, 0.75, 0.975} of the posterior are estimated at least with precision given, respectively, by {0.02, 0.05, 0.01, 0.05, 0.02} with probability 0.95 according to the Raftery-Lewis criterion (35) [Chapter 7].

Simulated data come from ARMAX models randomly obtained as follows. The polynomials  $B_k$ ,  $C$  and  $D := F_k$  entering Eq. 6 are generated by using **drmodel.m**: the first call defines  $B_1$  and  $D$ , the others the numerators of the remaining  $p$  rational transfer functions. The system is used and saved when the

following two requirements are satisfied. System poles must stay inside the circle of radius 0.95 while the signal-to-noise ratio must satisfy  $1 \leq \frac{\sum_{i=1}^p \|G_i\|_2^2}{\|H\|_2^2} \leq 10$  (recall that  $G_k(q) = \frac{B_k(q)}{D(q)}$ ,  $H(q) = \frac{C(q)}{D(q)}$ ), where  $\|G_i\|_2$ ,  $\|H\|_2$  indicate the  $\ell_2$  norms of the system impulse responses.

Finally, the reader is referred to **SI Appendix** for the proof of Proposition 1.

**Data, Materials, and Software Availability.** The code implementing the Full Bayes approach and the data are available at <https://www.dei.unipd.it/~giapi> under the voice Software. Previously published data were used for this work. Data are taken from the works: (9, 10, 57).

**ACKNOWLEDGMENTS.** Ljung's work was supported by the Swedish Research Council, under contract 2019-04956 and the Vinnova center LINKSIC. The paper was also partly funded by the project Proactive entitled Personalized whole-brain network models for neuroscience: modeling, inference and validation. Gianluigi Pillonetto would like to thank Giacomo Baggio, Alessandra Bertoldo and Alessandro Chiuso for helpful discussions on the topic.

Author affiliations: <sup>a</sup>Department of Information Engineering, University of Padova, 35131 Padova, Italy; and <sup>b</sup>Department of Electrical Engineering, Linköping University, S-581 83 Linköping, Sweden

Author contributions: G.P. and L.L. designed research; performed research; analyzed data; and wrote the paper.

1. L. Ljung, *System Identification - Theory for the User* (Prentice-Hall, Upper Saddle River, N.J., ed. 2, 1999).
2. T. Söderström, P. Stoica, *System Identification* (Prentice-Hall, 1989).
3. L. Zadeh, On the identification problem. *IRE Trans. Circuits Theory* **3**, 277-281 (1956).
4. G. Pillonetto, T. Chen, A. Chiuso, G. De Nicolao, L. Ljung, *Regularized System Identification* (Springer, 2022).
5. P. Van Overschee, B. De Moor, *Subspace Identification for Linear Systems: Theory - Implementation - Applications* (Springer-Verlag, 1996).
6. P. Hagmann *et al.*, Mapping the structural core of human cerebral cortex. *PLOS Biol.* **6**, 1-15 (2008).
7. R. Hickman *et al.*, Architecture and dynamics of the Jasmonic acid gene regulatory network. *Plant Cell* **29**, 2086-2105 (2017).
8. G. Pagani, M. Aiello, The power grid as a complex network: A survey. *Phys. A: Stat. Mech. Appl.* **392**, 2688-2700 (2013).
9. G. Prando *et al.*, Sparse DCM for whole-brain effective connectivity from resting-state fMRI data. *NeuroImage* **208**, 116367. (2020).
10. G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, L. Ljung, Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica* **50**, 657-682 (2014).
11. G. Pillonetto, G. De Nicolao, A new kernel-based approach for linear system identification. *Automatica* **46**, 81-93 (2010).
12. G. Pillonetto, A. Chiuso, G. De Nicolao, Prediction error identification of linear systems: A nonparametric Gaussian regression approach. *Automatica* **47**, 291-305 (2011).
13. T. Chen, H. Ohlsson, L. Ljung, On the estimation of transfer functions, regularizations and Gaussian processes - revisited. *Automatica* **48**, 1525-1535 (2012).
14. B. Schölkopf, A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, (Adaptive Computation and Machine Learning)* (MIT Press, 2001).
15. N. Aronszajn, Theory of reproducing kernels. *Trans. Am. Math. Soc.* **68**, 337-404 (1950).
16. S. Saitoh, *Theory of Reproducing Kernels and its Applications*, Pitman Research Notes in Mathematics Series (Longman Scientific and Technical, Harlow, 1988).
17. F. Dinuzzo, Kernels for linear time invariant system identification. *SIAM J. Control Optim.* **53**, 3299-3317 (2015).
18. M. Bisiacco, G. Pillonetto, On the mathematical foundations of stable RKHSs. *Automatica* (2020).
19. M. Bisiacco, G. Pillonetto, Kernel absolute summability is sufficient but not necessary for RKHS stability. *SIAM J. Control Optim.* (2020).
20. M. West, J. Harrison, *Bayesian Forecasting and Dynamic Models* (Springer-Verlag, Berlin, Heidelberg, ed. 2, 1997).
21. A. Gelman, J. Carlin, H. Stern, D. Rubin, *Bayesian Data Analysis* (Chapman and Hall/CRC, 2004).
22. G. Kimeldorf, G. Wahba, A correspondence between Bayesian estimation of stochastic processes and smoothing by splines. *Ann. Math. Statist.* **41**, 495-502 (1971).
23. A. Aravkin, B. Bell, J. Burke, G. Pillonetto, The connection between Bayesian estimation of a Gaussian random field and RKHS. *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 1518-1524 (2015).
24. C. Rasmussen, C. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, 2006).
25. G. Pillonetto, A. Chiuso, G. De Nicolao, "Regularized estimation of sums of exponentials in spaces generated by stable spline kernels" in *Proceedings of the IEEE American Control Conference, Baltimore, USA* (2010).
26. T. Chen *et al.*, Maximum entropy properties of discrete-time first-order stable spline kernel. *Automatica* **66**, 34-38 (2016).
27. E. Jaynes, On the rationale of maximum-entropy methods. *Proc. IEEE* **70**, 939-952 (1982).
28. B. Efron, C. Morris, Stein's estimation rule and its competitors-an empirical Bayes approach. *J. Am. Statist. Assoc.* **68**, 117-130 (1973).
29. J. S. Maritz, L. Lwin, *Empirical Bayes Method* (Chapman and Hall, 1989).
30. F. Carli, T. Chen, A. Chiuso, L. Ljung, G. Pillonetto, "On the estimation of hyperparameters for Bayesian system identification with exponential kernels" in *Proceedings of the IEEE Conference on Decision and Control (CDC 2012)* (2012).
31. A. Aravkin, J. Burke, A. Chiuso, G. Pillonetto, On the estimation of hyperparameters for empirical Bayes estimators: Maximum marginal likelihood vs minimum MSE. *IFAC Proc.* **45**, 125-130 (2012).
32. G. De Nicolao, G. Sparacino, C. Cobelli, Nonparametric input estimation in physiological systems: Problems, methods and case studies. *Automatica* **33**, 851-870 (1997).
33. D. MacKay, Bayesian interpolation. *Neural Comput.* **4**, 415-447 (1992).
34. T. Chen, L. Ljung, Implementation of algorithms for tuning parameters in regularized least squares problems in system identification. *Automatica* **49**, 2213-2220 (2013).
35. W. Gilks, S. Richardson, D. Spiegelhalter, *Markov Chain Monte Carlo in Practice* (Chapman and Hall, London, 1996).
36. A. Aravkin, J. Burke, A. Chiuso, G. Pillonetto, Convex vs non-convex estimators for regression and sparse estimation: The mean squared error properties of ARD and Glasso. *J. Mach. Learn. Res.* **15**, 217-252 (2014).
37. B. Mu, T. Chen, L. Ljung, "Asymptotic properties of hyperparameter estimators by using cross-validations for regularized system identification" in *2018 IEEE Conference on Decision and Control (CDC)* (2018), pp. 644-649.
38. B. Mu, T. Chen, L. Ljung, On asymptotic properties of hyperparameter estimators for kernel-based regularization methods. *Automatica* **94**, 381-395 (2018).
39. M. Gu, X. Wang, J. Berger, Robust Gaussian stochastic process emulation. *Ann. Stat.* **46**, 3038-3066 (2018).
40. G. Pillonetto, A. Chiuso, Tuning complexity in regularized kernel-based regression and linear system identification: The robustness of the marginal likelihood estimator. *Automatica* **58**, 106-117 (2015).
41. D. Wipf, B. Rao, Sparse Bayesian learning for basis selection. *IEEE Trans. Signal Process.* **52**, 2153-2164 (2004).
42. B. Ninness, S. Henriksen, Bayesian system identification via Markov Chain Monte Carlo techniques. *Automatica* **46**, 40-51 (2010).
43. T. Schon *et al.*, Sequential Monte Carlo methods for system identification. *IFAC-PapersOnLine* **48**, 775-786 (2015).
44. J. Hendriks, A. Wills, B. Ninness, J. Dahlin, Practical Bayesian system identification using Hamiltonian Monte Carlo (2021). <http://arxiv.org/abs/2011.04117>.
45. P. Magni, R. Bellazzi, G. De Nicolao, Bayesian function learning using MCMC methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 1319-1331 (1998).
46. M. Gu, H. Li, Gaussian orthogonal latent factor processes for large incomplete matrices of correlated data. *Bayesian Anal.* **17**, 1219-1244 (2022).
47. M. Gu, X. Liu, X. Fang, S. Tang, Scalable marginalization of latent variables for correlated data with applications to learning particle interaction kernels. *New Engl. J. Stat. Data Sci.* (2022).
48. P. Van den Hof, A. Danks, P. Heuberger, X. Bombois, Identification of dynamic models in complex networks with prediction error methods: Basic methods for consistent module estimates. *Automatica* **49**, 2994-3006 (2013).
49. K. Ramaswamy, G. Bottegal, P. J. Van den Hof, Learning linear models in a dynamic network using regularized kernel-based methods. *Automatica* **129**, 109591. (2021).
50. Z. Yue, J. Thunberg, W. Pan, L. Ljung, J. Goncalves, Dynamic network reconstruction from heterogeneous datasets. *Automatica* **123**, 109339. (2021).
51. G. Pillonetto, A. Yazdani, Sparse estimation in linear dynamic networks using the stable spline horseshoe prior. *Automatica* **146**, 110666. (2022).
52. H. Jeffreys, An invariant form for the prior probability in estimation problems. *Proc. R. Soc. Lond.* **186**, 453-461 (1946).

53. J. Berger, V. De Oliveira, B. Sansó, Objective Bayesian analysis of spatially correlated data. *J. Am. Stat. Assoc.* **96**, 1361–1374 (2001).
54. J. Berger, J. Bernardo, D. Sun, Overall objective priors. *Bayesian Anal.* **10**, 189–221 (2015).
55. G. Wahba, *Spline Models for Observational Data* (SIAM, Philadelphia, 1990).
56. T. Chen, M. S. Andersen, L. Ljung, A. Chiuso, G. Pillonetto, System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques. *IEEE Trans. Autom. Control* **59**, 2933–2945 (2014).
57. G. Pillonetto, T. Chen, A. Chiuso, G. De Nicolao, L. Ljung, Regularized linear system identification using atomic, nuclear and kernel-based norms: The role of the stability constraint. *Automatica* **69**, 137–149 (2016).
58. R. Buxton, E. Wong, L. Frank, Dynamics of blood flow and oxygenation changes during brain activation: The balloon model. *Magn. Reson. Med.* **39**, 855–64 (2018).
59. J. Kim, D. Bassett, Linear dynamics and control of brain networks. *Neural Eng.* 497–518 (2020).
60. G. Baggio, D. Bassett, F. Pasqualetti, Data-driven control of complex networks. *Nat. Commun.* (2021).
61. D. Materassi, G. Innocenti, Topological identification in networks of dynamical systems. *IEEE Trans. Autom. Control* **55**, 1860–1871 (2010).
62. A. Chiuso, G. Pillonetto, A Bayesian approach to sparse dynamic network identification. *Automatica* **48**, 1553–1565 (2012).
63. K. Friston, L. Harrison, W. Penny, Dynamic causal modelling. *Neuroimage* **19**, 1273–1302 (2003).
64. K. Friston, J. Kahan, B. Biswal, A. Razi, A DCM for resting state fMRI. *Neuroimage* **94**, 396–407 (2014).
65. A. Savva, G. Mitsis, G. Matsopoulos, Assessment of dynamic functional connectivity in resting-state fMRI using the sliding window technique. *Brain Behav.* **9** (2019).
66. E. Camacho, C. Bordons, *Model Predictive Control, Advanced Textbooks in Control and Signal Processing* (Springer Verlag, 2004).
67. M. Yudong *et al.*, "Model predictive control for the operation of building cooling systems" in *American Control Conference* (2010), pp. 5106–5111.
68. S. Prvara, J. Siroky, L. Ferkl, J. Cigler, Predicting hourly building energy use: The great energy predictor shootout: Overview and discussion of results. *Energy Buil.* **43**, 45–48 (2011).
69. J. Schoukens, L. Ljung, Nonlinear system identification - a user-oriented roadmap. *IEEE Control Syst. Mag.* **39**, 28–99 (2019).
70. G. Pillonetto, A. Chiuso, M. H. Quang, A new kernel-based approach for nonlinear system identification. *IEEE Trans. Autom. Control* **56**, 2825–2840 (2011).
71. G. Pillonetto, System identification using kernel-based regularization: New insights on stability and consistency issues. *Automatica* **93**, 321–332 (2018).