System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques

Tianshi Chen, *Member, IEEE*, Martin S. Andersen, Lennart Ljung, *Life Fellow, IEEE*, Alessandro Chiuso, *Senior Member, IEEE*, Gianluigi Pillonetto, *Member, IEEE*,

***Abstract*—Model estimation and structure detection with short data records are two issues that receive increasing interests in System Identification. In this paper, a multiple kernel-based regularization method is proposed to handle those issues. Multiple kernels are conic combinations of fixed kernels suitable for impulse response estimation, and equip the kernel-based regularization method with three features. First, multiple kernels can better capture complicated dynamics than single kernels. Second, estimation of their weights by maximizing the marginal likelihood favors sparse optimal weights, which enables this method to tackle various structure detection problems, e.g., the sparse dynamic network identification and the segmentation of linear systems. Third, the marginal likelihood maximization problem is a difference of convex programming problem. It is possible to find a locally optimal solution efficiently by using a majorization minimization algorithm and an interior point method where the cost of a single interior-point iteration grows linearly in the number of fixed kernels. Monte Carlo simulations show that the locally optimal solutions lead to good performance, regardless of the initialization.**

***Index Terms*—System identification, regularization, kernel, convex optimization, sparsity, structure detection.**

I. Introduction

SYSTEM Identification is a mature field, see e.g., the textbooks [1]–[3]. However, the increasingly complex engineering systems pose new challenges in terms of efficiency, robustness, reliability and autonomy. We are faced with many emerging issues in System Identification including model estimation and structure detection with short data records:

Model estimation. The standard approach to System Identification is the maximum likelihood/prediction error method (ML/PEM), e.g., [1]. It has optimal asymptotic properties in the number of data points: if the model structure contains the true system, the estimated model will converge to the true system with smallest possible variance. However, available data records are often short in practice due to cost and/or time reasons. As shown by extensive simulations in [4]–[6], ML/PEM equipped with classical model structure selection techniques sometimes fails to get model estimates with good accuracy and robustness for short and noisy data records.

Structure detection. Structural constraints widely exist in engineering systems. In networked and decentralized systems, certain inputs usually influence only certain outputs. In piecewise affine systems, each data point must be associated to the most suitable submodel. They are often tackled, see e.g., [7], in a ML or related framework by using ARX model and LASSO [8], group LASSO [9] techniques. However, for short data records, possible high variance of ARX model may deteriorate the detection accuracy. Moreover, there are other sparsity techniques, e.g., sparse Bayesian learning (SBL) [10], [11], which can produce more sparse solutions with also more favorable properties in terms of mean square error (MSE), see e.g., [12]–[14].

A new approach, which has been shown particularly useful for model estimation with short data records, is the kernel-based regularization method (KRM) introduced in [4] and further studied in [5], [6]. Its performance depends on both kernel structure design, i.e., parameterization of the kernel by some parameters often called hyper-parameters, and hyper-parameter estimation. There are several ways for the hyper-parameter estimation, e.g. [15], [16]. So far, the most effective one is to embed the regularization in Bayesian framework and invoke the empirical Bayes method, i.e., the marginal likelihood maximization method. This method embodies an *automatic Occam's razor (parsimonious) principle*, i.e., trade-off between data fit and model complexity [10], [15, p. 110], which is an important reason why KRM outperforms ML/PEM equipped with the classical model structure selection techniques in dealing with the bias-variance tradeoff for short and noisy data records. Since [4], several kernel structures have been introduced [5], [6], [17], [18]. However, these kernels could be improved if the true system has complicated dynamics, e.g., with several widely spread time constants. An outstanding question is how to parameterize the kernel with flexible structure so that complicated dynamics can be better captured?

Interestingly, KRM in [6] also has close connection with SBL [10], [11], which is a Bayesian method for finding sparse solutions and has advantages over LASSO and Group LASSO

T. Chen (corresponding author) and L. Ljung are with the Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, 58183, Sweden. e-mail: tschen@isy.liu.se, ljung@isy.liu.se. M. S. Andersen is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark. e-mail: mskan@dtu.dk. Alessandro Chiuso and Gianluigi Pillonetto are with the Department of Information Engineering, University of Padova. e-mail: chiuso@dei.unipd.it,giapi@dei.unipd.it.

The authors thank the reviewers and guest editors for their constructive comments. This research has been partially supported by the Linnaeus Center CADICS, funded by the Swedish Research Council, and the ERC advanced grant LEARN, no. 267381, funded by the European Research Council as well as by the MIUR FIRB project "Learning meets time" (RBFR12M3AC) and European Community's Seventh Framework Programme [FP7/2007-2013] under agreement no. 257462 HYCON2 Network of excellence.

in terms of sparsity property [12], [13] and in terms of MSE [14]. In fact, if the kernel in [6] is diagonal and has all diagonal elements as hyper-parameters, KRM in [6] becomes SBL for basis selection [12]. It finds sparse solutions in the hyper-parameter space, which in turn leads to sparse solutions in the parameter (hypothesis) space. Noticing this connection, a natural question is whether it is possible to incorporate SBL's feature of favoring sparsity into KRM and tackle accordingly structure detection problems in System Identification [7]?

Both questions aforementioned are related to kernel structure design. Indeed, we are looking for kernel structures that can both capture complicated dynamics and induce sparse hyper-parameters (and sparse hypotheses in the end), but is there any other concern that should be considered? Since the marginal likelihood maximization problem is non-convex and often has no closed-form solution, in our opinion, one such concern is if the designed kernel structure can bring the marginal likelihood maximization problem certain structures so that a locally optimal solution can be found efficiently.

In this paper, we aim to address the three questions raised above. Noticing the superposition property of linear systems, it is natural to propose using the multiple kernel, which is a conic combination of suitable fixed kernels and has the combination coefficients as hyper-parameters. The fixed kernels can be instances of existing single kernels [5], [6], [17], [18] and can also be constructed based on model estimates which can be either data-driven or data-free. Due to their flexible structures, multiple kernels can better capture complicated dynamics than single kernels. What's more, the marginal likelihood maximization problem with multiple kernel favors sparse hyper-parameters. This feature enables this multiple KRM (MKRM) to tackle various structure detection problems. For illustration, the sparse dynamic network identification [19] and the segmentation problems of linear systems [20] will be studied here. For both model estimation and structure detection, MKRM reduces to a marginal likelihood maximization problem. The multiple kernel brings the problem a special structure that it is a difference of convex programming (DCP) problem [21], [22]. Its locally optimal solution can be found efficiently using sequential convex optimization techniques. In particular, we use a majorization minimization (MM) algorithm [23], [24] and an interior-point method, where the cost of a single interior-point iteration grows linearly in the number of fixed kernels. Monte Carlo simulations show that the locally optimal solutions lead to good performance, *regardless of the initialization*, which is a practical advantage over ML/PEM and KRM with nonlinearly parameterized kernels where the initialization is critical and tricky.

The remaining parts of this paper is organized as follows. MKRM is proposed in Section II where it is also shown that the marginal likelihood maximization with multiple kernel is a DCP problem. By exploiting this structure, its locally optimal solution is found in Section III by using an MM algorithm and an interior point method. In Section IV, it is further shown that the marginal likelihood maximization with multiple kernel favors sparse hyper-parameters. This feature is then used to study the sparse dynamic network identification and the segmentation problems of linear systems. To illustrate the effectiveness of the proposed method, three sets of simulations, two of which are Monte Carlo ones, are considered in Section V. We finally conclude this paper in Section VI.

II. Model estimation with multiple kernel-based regularization

A. Problem statement

Consider a single-input-single-output (SISO) linear casual and stable system

$$y(t) = G_0(q)u(t) + v(t), \tag{1}$$

where t is the time index (the sampling interval is assumed to be one time unit), q is the shift operator, meaning $qu(t) = u(t+1)$, $y(t), u(t)$ and $v(t)$ are the output, input and disturbance at time t, respectively. The disturbance $v(t)$ is modeled as a white noise with mean zero and variance σ^2, independent of $u(t)$; see Remark 2.5 for discussions about the case where $v(t)$ is modeled as a filtered white noise. The transfer function $G_0(q)$ can be written as $G_0(q) = \sum_{k=1}^{\infty} g_k^0 q^{-k}$, where the coefficients $g_k^0, k = 1, \cdots, \infty$, form the impulse response of $G_0(q)$. Given a data record $\{u(t), y(t)\}_{t=1}^N$, the goal is to find an estimate of $G_0(q)$, or equivalently, an estimate of the impulse response of $G_0(q)$ that is as good as possible.

B. Regularized FIR model estimation

Consider system (1). Since the impulse response of a linear system decays exponentially, it is often enough to truncate the infinite impulse response at a certain order and estimate an FIR (finite impulse response) model

$$G(q,\theta) = \sum_{k=1}^{n} g_k q^{-k}, \quad \theta = \begin{bmatrix} g_1 & g_2 & \cdots & g_n \end{bmatrix}^T. \tag{2}$$

The model of system (1) can then be written as

$$y(t) = \phi(t)^T \theta + v(t), \qquad t = n+1, \cdots, N, \tag{3}$$

with $\phi(t)^T = [u(t-1) \ \cdots \ u(t-n)]$, which can be further written in a more compact form

$$Y_N = \Phi_N^T \theta + V_N. \tag{4}$$

The ith row of $Y_N, V_N \in \mathbb{R}^{N-n}$ and $\Phi_N^T \in \mathbb{R}^{(N-n)\times n}$ are $y(n+i), v(n+i)$ and $\phi(n+i)^T$, respectively. For $t = 1, \cdots, n$, $y(t)$ depends the unknown $u(0), \cdots, u(t-n)$, which can be handled in different ways, see [1, p. 320]. Like [6], the non-windowed method is used here, i.e., $y(t)$, $t = 1, \cdots, n$ are not used. Then the regularized least squares estimate $\hat{\theta}_N^R$ of θ is

$$\hat{\theta}_N^R = \arg\min_{\theta} \|Y_N - \Phi_N^T \theta\|_2^2 + \sigma^2 \theta^T P^{-1} \theta \tag{5a}$$

$$= P\Phi_N(\Phi_N^T P \Phi_N + \sigma^2 I_{N-n})^{-1} Y_N, \tag{5b}$$

where I_{N-n} denotes the $N-n$ dimensional identity matrix and P is positive semi-definite (denoted by $P \succeq \mathbf{0}$) and often called kernel (matrix) in Machine Learning [15] and Bayesian Framework [29]. If P is positive definite, it is denoted by $P \succ \mathbf{0}$ below, where $\mathbf{0}$ denotes a zero matrix with suitable dimension which can be judged from the context.

Remark 2.1: When P is singular, (5a) is not well-defined. In this case, consider the singular value decomposition of P:

$P = [U_1\ U_2]\mathrm{diag}(\Lambda_P, \mathbf{0})[U_1\ U_2]^T$ where Λ_P is a diagonal matrix with diagonal elements being strictly positive singular values of P, $[U_1\ U_2]$ is an orthogonal matrix with U_1 having the same number of columns as Λ_P, and $\mathrm{diag}(\Lambda_P, \mathbf{0})$ is a block-diagonal matrix with Λ_P and $\mathbf{0}$ on the main diagonal. Then (5a) should be interpreted as

$$\hat{\theta}_N^R = \arg\min_{\theta} \|Y_N - \Phi_N^T\theta\|_2^2 + \sigma^2\theta^T U_1\Lambda_P^{-1}U_1^T\theta, \text{s.t. } U_2^T\theta = 0 \tag{6}$$

It is easy to verify that (5b) is still the optimal solution of (6). For convenience, we will still use (5a) in the sequel and refer to (6) for its rigorous meaning when P is singular.

Assume $g_k^0 = 0, k > n$ so that the true system (1) is described by an FIR model, and denote the true impulse response coefficients by $\theta_0 = \begin{bmatrix} g_1^0 & g_2^0 & \cdots & g_n^0 \end{bmatrix}^T$. Then we have the following convergence result for $\hat{\theta}_N^R$.

Theorem 2.1: Assume that $u(t)$ is deterministic, $v(t)$ is i.i.d. with mean 0 and variance σ^2, $\Phi_N\Phi_N^T/N \to \Omega$ as $N \to \infty$ where Ω is positive definite, and $\Phi_N V_N/N \to \mathbf{0}$ with probability one as $N \to \infty$. If θ_0 can be represented as a linear combination of eigenvectors of $P\Omega$, then $\hat{\theta}_N^R \to \theta_0$ with probability one as $N \to \infty$.

Proof: The proof is straightforward and omitted due to the limitation of space.

Remark 2.2: The kernel P brings the regularized FIR model estimate $\hat{\theta}_N^R$ a special structure. Since $\Phi_N(\Phi_N^T P\Phi_N + \sigma^2 I_{N-n})^{-1}Y_N$ is a column vector, $\hat{\theta}_N^R$ is a linear combination of the column vectors of P. It is preferable to have the column space of P include θ_0. From Theorem 2.1, if Ω is nonsingular and P is nonsingular or $P = c\theta_0\theta_0^T$ for $c > 0$, θ_0 can be represented as a linear combination of the eigenvectors of $P\Omega$ and thus $\hat{\theta}_N^R$ is asymptotically consistent.

C. Multiple kernel

The design of P consists of two parts: kernel structure design, i.e., parameterization of P by some parameters called hyper-parameters, and hyper-parameter estimation for a kernel structure. Many efforts have been spent on designing kernel structures and several kernels have been introduced in [4]–[6], [17], [18], e.g., the stable spline (SS), the tuned/correlated (TC) and the diagonal/correlated (DC) kernels:

$$\text{SS} \qquad cP_{k,j}^{ss}(\beta) = c\begin{cases} \frac{\lambda^{2k}}{2}(\lambda^j - \frac{\lambda^k}{3}), k \geq j \\ \frac{\lambda^{2j}}{2}(\lambda^k - \frac{\lambda^j}{3}), k < j \end{cases}, \ \beta = \lambda \tag{7a}$$

$$\text{TC} \qquad cP_{k,j}^{tc}(\beta) = c\min(\lambda^k, \lambda^j), \ \beta = \lambda \tag{7b}$$

$$\text{DC} \qquad cP_{k,j}^{dc}(\beta) = c\lambda^{(k+j)/2}\rho^{|k-j|}, \ \beta = [\lambda \quad \rho]^T \tag{7c}$$

where c, β are hyper-parameters with $c \geq 0$, $0 \leq \lambda < 1$, $|\rho| \leq 1$, and the subscript k, j denotes the (k, j) element of a matrix.

However, those single kernels $cP(\beta)$ in (7) could be improved to better capture complicated dynamics. For example, consider the case $\theta_0 = \theta_1 + \theta_2$ where $\theta_1, \theta_2 \in \mathbb{R}^n$ are two FIRs that have very different dynamics in terms of e.g., decay rate and magnitude. Instead of $cP(\beta)$, better impulse response estimate $\hat{\theta}_N^R$ can often be obtained using $c_1P(\beta_1) + c_2P(\beta_2)$, but the initialization and optimization becomes more tricky for the associated hyper-parameter estimation problem. Interestingly, the domain of β is compact, so if there is no knowledge about the estimate of $\beta_i, i = 1, 2$, it is natural to introduce a grid of points $\beta_1, \ldots, \beta_m$ over the domain of β and use the kernel $\sum_{i=1}^m c_iP(\beta_i)$ with $c_1, \ldots, c_m$ being the hyper-parameters instead. From this observation and the supposition property of linear systems that impulse response of a linear system is the sum of impulse responses of its partial fraction expansion, it is natural to propose using the multiple kernel

$$P(\alpha) = \sum_{i=1}^m c_iP_i, \quad \alpha = \begin{bmatrix} c_1, \cdots, c_m \end{bmatrix}^T, \tag{8}$$

where $c_i \geq 0$, $P_i \succeq \mathbf{0}$ and $P_i \neq \mathbf{0}$, $i = 1, \cdots, m$, are fixed kernels. The fixed kernels P_i can be constructed in different ways. In what follows, we mainly consider the way to construct P_i as instances of single kernels (7), but in Section V-A we will briefly discuss another way.

Example 2.1: We illustrate the advantage of using multiple kernel by a simple example:

$$G_0(q) = z_1q^{-1}(1 - p_1q^{-1})^{-1} + z_2q^{-1}(1 - p_2q^{-1})^{-1}, \tag{9}$$

where $z_1 = 1, z_2 = -50$ and $p_i, i = 1, 2$ are generated as `p1 = rand(1)/2+0.5` and `p2 = sign(randn(1))*rand(1)/2` in MATLAB. Example (9) contains two distinct modes: the fast one dominates the dynamics in the initial phase and the slow one dominates afterwards. Here, 1000 instances of (9) and associated data sets are generated. A multiple kernel (8) is constructed with 20 fixed kernels obtained by evaluating (7b) on the grid with $c = 1$, $\lambda = 0.05 : 0.05 : 0.95, 0.98$. This multiple kernel (denoted by TC-M) is compared with the single kernels (7). The simulation result in terms of model fit (26) is shown on the left panel of Fig. 1. The advantage of using multiple kernel is quite clear. As can be seen from the right panel of Fig. 1, the single kernels try to capture the fast mode in the initial phase so that they, unlike the multiple kernel, do not have extra flexibility to well capture the slow mode.

D. Hyper-parameter estimation

Given a multiple kernel $P(\alpha)$, there exist several ways to estimate the hyper-parameter α. Currently, the most effective one is to embed the regularization term $\theta^T P^{-1}\theta$ in (5a) in Bayesian framework and estimate α by maximizing the marginal likelihood.

Assume $v(t)$ in (1) is Gaussian distributed, independent of the input, and

$$\theta \sim \mathcal{N}(\theta^{ap}, P(\alpha)), \qquad \theta^{ap} = \mathbf{0}, \tag{10}$$

where θ^{ap} is the prior mean and $P(\alpha)$ is the prior covariance. Note that θ^{ap} can be nonzero, see [6, Section 4.2]. It is easy to show that the maximum a posteriori (MAP) estimation problem $\arg\max_\theta p(\theta|Y_N)$ is equivalent to (5a). The marginal likelihood, i.e., Y_N conditioned on α is Gaussian distributed as $p(Y_N|\alpha) = \mathcal{N}(\mathbf{0}, \Phi_N^T P(\alpha)\Phi_N + \sigma^2 I_{N-n})$. The marginal likelihood maximization method $\arg\max_{\alpha \geq \mathbf{0}} p(Y_N|\alpha, \sigma^2)$ to estimate α is equivalent to

$$\hat{\alpha} = \arg\min_{\alpha \geq \mathbf{0}} Y_N^T\Sigma(\alpha, \sigma^2)^{-1}Y_N + \log\det\Sigma(\alpha, \sigma^2), \tag{11}$$

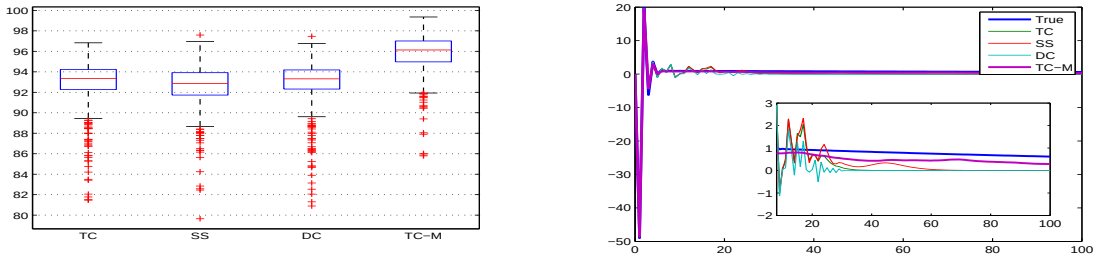

Fig. 1. Box-plots of the 1000 model fits (left) and the estimated impulse responses for one instance (right).

where $\Sigma(\alpha,\sigma^2)=\Phi_N^T P(\alpha)\Phi_N+\sigma^2 I_{N-n}$.

Although (11) is non-convex, the multiple kernel $P(\alpha)$ renders (11) a special structure. Note from e.g. [30] that both $Y_N^T\Sigma(\alpha,\sigma^2)^{-1}Y_N$ and $-\log\det\Sigma(\alpha,\sigma^2)$ are convex in $\Sigma(\alpha,\sigma^2)\succ\mathbf{0}$ and $\Sigma(\alpha,\sigma^2)$ is affine in α and σ^2. Therefore both $Y_N^T\Sigma(\alpha,\sigma^2)^{-1}Y_N$ and $-\log\det\Sigma(\alpha,\sigma^2)$ are convex in $\alpha\geq\mathbf{0}$ and $\sigma^2>0$, respectively. So the objective function of (11) is a difference of two convex functions with respect to $\alpha\geq\mathbf{0}$ and $\sigma^2>0$, and thus (11) is a difference of convex programming (DCP) problem [21], [22]. It will be shown in Section III that a stationary point of (11) can be found efficiently using sequential convex optimization techniques.

Remark 2.3: The noise variance σ^2 is not known and needs to be estimated from the data. As suggested in [1], [31], a simple and effective way is to estimate an ARX model [4], [5] or an FIR model [6] with least squares and use the sample variance as the estimate of σ^2. An alternative way is to treat σ^2 as an additional "hyper-parameter" and estimate it together with α by solving (11), see e.g., [10]. All arguments below (with minor changes) still hold with σ^2 as an optimization argument in (11). At least for the test data bank in Section V-A, the alternative way seems a better choice for MKRM.

Remark 2.4: If $m=n$ and $P_i=e_ie_i^T$, $i=1,\cdots,n$ in (8), where $e_i\in\mathbb{R}^n$ has its ith element equal to 1 and all other elements equal to zero, MKRM becomes SBL for basis selection [12]. It favors sparse α in the hyper-parameter space and in turn leads to sparse θ in the parameter space. This observation prompts us to ask if MKRM has the same feature, as SBL for basis selection, of favoring sparse α in the more general multiple kernel (8). The answer to this question is affirmative and will be discussed in Section IV, see also [27, Thm.1].

Remark 2.5: Consider the case where $v(t)$ in (1) is a filtered white noise, i.e., $v(t)=H_0(q)e(t)$. Here $H_0(q)$ is unknown, both stable and inversely stable [1] with unit static gain, and $e(t)$ is a white noise with mean zero and variance σ^2. Now our goal is to estimate $G_0(q)$ and $H_0(q)$ as well as possible. Recall that system (1) with $v(t)=H_0(q)e(t)$ can be well approximated (see [32]) by a high order ARX model $y(t)=A_n(q)^{-1}B_n(q)u(t)+A_n(q)^{-1}e(t)$ with $A_n(q)=1+a_1q^{-1}+\ldots+a_nq^{-n}$, $B_n(q)=b_1q^{-1}+\ldots+b_nq^{-n}$, which can be written as a linear regression $y(t)=\phi_y^T(t)\theta_a+\phi_u^T(t)\theta_b+e(t)$ where $\theta_a=[a_1,\ldots,a_n]^T$, $\theta_b=[b_1,\ldots,b_n]^T$, and $\phi_y(t),\phi_u(t)$ are defined in an obvious way. Note that for large n, θ_a,θ_b can be interpreted as the two FIRs for the one-step-ahead predictor of system (1) with $v(t)=H_0(q)e(t)$ from y and u, respectively, see [1]. So the ARX model estimation problem becomes an FIR model estimation problem with two inputs and the same idea of regularization can be applied. Similar to (5a),

$$\hat\theta_a,\hat\theta_b=\arg\min_{\theta_a,\theta_b}\sum_{t=n+1}^{N}(y(t)-\phi_y^T(t)\theta_a-\phi_u^T(t)\theta_b)^2+\sigma^2\begin{bmatrix}\theta_a & \theta_b\end{bmatrix}Q^{-1}\begin{bmatrix}\theta_a & \theta_b\end{bmatrix}^T.$$

It is intuitive to partition Q as a block-diagonal matrix $Q(\alpha_a,\alpha_b)=\mathrm{diag}(P(\alpha_a),P(\alpha_b))$. The hyper-parameters α_a,α_b are still estimated by maximizing the marginal likelihood.

Remark 2.6: The idea of using multiple kernel (8) has appeared in machine learning [25], [26] and neuroimaging [27]. Multiple kernel learning (MKL) [25], [26] is one such method and has also been used recently to handle linear system identification problems in [28]. It can be shown that, see e.g. [14] for derivations for group variable selection case, MKL also reduces to an estimation problem of α:

$$\hat\alpha=\arg\min_{\alpha\geq\mathbf{0}}Y_N^T\Sigma(\alpha,\sigma^2)^{-1}Y_N+\gamma\mathbf{1}^T\alpha, \tag{12}$$

where $\mathbf{1}=[1\ 1\ \cdots\ 1]^T$. Clearly, (12) is much easier to solve than (11) since (12) is convex. However, there is a price to pay for that. The comparison between (11) and (12) shows that their difference lies in the second term: $\log\det(\Phi_N^TP(\alpha)\Phi_N+\sigma^2I_{N-n})$ is replaced by $\gamma\mathbf{1}^T\alpha$. So MKL actually solves a suboptimal marginal likelihood maximization problem. Such an approximation often results in less accurate and less robust model estimates for model estimation problems, and in general tends to produce less sparse solutions with also less favorable properties in terms of MSE for sparsity problems, see e.g. [14]. The hyper-parameter estimation problem in [27] is also solved by maximizing the marginal likelihood but the algorithm and implementation are different from the one in this paper. Due to the space limitation, detailed comparison cannot be given here but in another paper.

III. Negative Log Marginal Likelihood Minimization with Multiple Kernel Using Sequential Convex Optimization Techniques

The hyper-parameter estimation problem (11) can be put into the following form

$$\underset{x\geq0}{\text{minimize}}\ Y^T\left(\sum_{i=1}^{p}x_iB_iB_i^T+\sigma^2I_{n_o}\right)^{-1}Y+\log\det\left(\sum_{i=1}^{p}x_iB_iB_i^T+\sigma^2I_{n_o}\right),x=\begin{bmatrix}x_1 & \ldots & x_p\end{bmatrix}^T, \tag{13}$$

where $Y\in\mathbb{R}^{n_o}$, $B_i\in\mathbb{R}^{n_o\times n_i}$ and $B_i\neq\mathbf{0}$, and $n_o,n_i,i=1,\cdots,p$, are positive integers. Since $P_i\succeq\mathbf{0}$, it can be factorized as $P_i=$

$L_iL_i^T$, where $L_i \in \mathbb{R}^{n\times n_i}$ with n_i being a positive integer. For example, for $P_i \succ \mathbf{0}$, L_i can be its Cholesky factorization and $n_i = n$. So (11) can be put into the form of (13) with $p = m$, $n_o = N - n$, $x_i = c_i$, $i = 1,\cdots,p$, $Y = Y_N$ and $B_i = \Phi_N^T L_i$, $i = 1,\cdots,p$. In what follows, (13) is referred to as the *negative log marginal likelihood minimization with multiple kernel*. This is because for both model estimation and structure detection, see Section IV, the associated negative log marginal likelihood minimization problems can all be put into the form of (13). In general, Y and x have the interpretation of measurement output and hyper-parameter, respectively, and B_i contains the information of the measurement input and the fixed kernel P_i in (8). Obviously, (13) is still a DCP problem. Now, we consider how to tackle (13) by exploiting its DC structure and using sequential convex optimization techniques.

A. Sequential convex optimization techniques: majorization minimization (MM) algorithms

There are a couple of sequential convex optimization techniques that can be used to tackle DCP problems. One of them is the so-called MM algorithm [23], [24] and its main idea is to yield an iterative scheme for $\text{minimize}_{x\in C} f(x)$ with $C \subseteq \mathbb{R}^p$ where each iteration consists of minimizing a so-called majorization function $\bar{f}(x,x^{(k)})$ of $f(x)$ at $x^{(k)} \in C$:

$$x^{(k+1)} = \arg\min_{x\in C} \bar{f}(x,x^{(k)}), \tag{14}$$

where $\bar{f} : C\times C \to \mathbb{R}$ satisfies $\bar{f}(x,x) = f(x)$ for $x \in C$ and $f(x) \le \bar{f}(x,z)$ for $x,z \in C$. Clearly, (14) yields a descent algorithm. Construction of a suitable majorization function is a key step for MM algorithms. For DCP problems $\text{minimize}_{x\in C} f(x)$ where $f(x) = g(x) - h(x)$, $g,h : C \to \mathbb{R}$ are convex and differentiable functions with C being a convex set in $\mathbb{R}^p$, there are many ways to construct the majorization function [24]. The simplest one is the so-called linear majorization or majorization via "supporting hyperplane" [24], i.e.,

$$\bar{f}(x,x^{(k)}) = g(x) - h(x^{(k)}) - \nabla h(x^{(k)})^T (x - x^{(k)}). \tag{15}$$

For this particular choice of majorization function, the MM algorithm (14) is also referred to as "sequential convex optimization" or "the convex concave procedure" (CCCP) [23].

Remark 3.1: The so-called simplified difference of convex functions algorithm (DCA) [21], [22] is another sequential convex optimization technique that can be used to tackle DCP problems. Simplified DCA is a primal–dual method that alternates between majorization minimization updates based on the problem $\inf_x\{g(x) - h(x)\}$ and its Fenchel–Rockafellar dual. For differentiable $f(x)$, the CCCP algorithm [23] is equivalent to a primal-only variant of the simplified DCA, which is a special case of MM algorithm (with linear majorization).

B. MM algorithms to the negative log marginal likelihood minimization with multiple kernel

From now on, we identify $f(x)$ as the objective function of (13) , $C = \{x \in \mathbb{R}^p | x \ge \mathbf{0}\}$ and $f(x) = g(x) - h(x)$ where

$$g(x) = Y^T\Sigma(x)^{-1}Y, \quad h(x) = -\log\det\Sigma(x),$$
$$\Sigma(x) = \sum_{i=1}^{p} x_i B_i B_i^T + \sigma^2 I_{n_o}. \tag{16}$$

Algorithm 3.1: The MM algorithm to the problem (13) can be summarized as follows: Set $x^{(0)}$, $k = 0$ and then go to the following iterative steps:

1) Compute the gradient $\nabla h(x^{(k)})$ according to $\nabla_{x_i} h(x) = -\text{Tr}\left(\Sigma(x)^{-1}\frac{\partial\Sigma(x)}{\partial x_i}\right)$, $i = 1,\cdots,p$, and then solve the convex optimization problem (14) and (15) to obtain $x^{(k+1)}$.
2) Check if the optimality condition is satisfied. If satisfied, stop. If otherwise, set $k = k+1$ and go to step 1).

The convergence of MM algorithms to a stationary point (the point satisfies the Karush-Kuhn-Tucker (KKT) conditions, see e.g., [30]) has been discussed in e.g., [33]. For the MM Algorithm 3.1, [33, Thm. 4] can be employed to show the convergence. In the following, we show the four assumptions of [33, Thm. 4] are satisfied for the MM algorithm 3.1. First, both $g(x)$ and $h(x)$ are real-valued differentiable convex functions. Second, $\nabla h(x)$ is obviously continuous. Third, for any $x \ge \mathbf{0}$, the set $H(x) = \{z | f(z) \le f(x), z \ge \mathbf{0}\}$ is indeed bounded. This is because for any $x \ge \mathbf{0}$ and $x \ne \mathbf{0}$, $\lim_{t\to+\infty}\left(Y^T\Sigma(tx)^{-1}Y + \log\det\Sigma(tx)\right) \to +\infty$. Fourth, there is no equality constraint involved and moreover, the inequality constraint $x \ge \mathbf{0}$ leads to $c_i(x) = -x_i$, $i = 1,\cdots,p$ in [33, eq. (1)], which are real-valued convex functions. Therefore, the MM Algorithm 3.1 converges to a stationary point of (13).

Remark 3.2: In [34], CCCP algorithm was used to solve the marginal likelihood maximization problem for basis selection [12]. Here, we consider MM algorithm instead of simplified DCA to tackle (13) primarily because the simplified DCA involves the conjugate function $h^*(y) = \sup_x\{y^Tx + \log\det\Sigma(x)\}$, which has no closed-form solution and thus is expensive to evaluate. There are also two secondary reasons. First, for a given DCP problem, different MM algorithms can be easily derived by employing different majorization functions. Second, noting that MM algorithms are not widely known as its special case expectation maximization (EM) algorithms [24] in System Identification, (13) shows that they can be alternative choices for parameter estimation problems in System Identification. In this regard, it is also interesting to note [35], which minimizes a convex upper bound of a nonconvex objective function for a nonlinear state-space model identification problem. If the procedure in [35] is done in a sequential way, it will be inline with the idea of handling DCP problems with sequential convex optimization techniques.

C. An efficient and accurate implementation

It is possible to solve each iteration (14) using a (fast) projected gradient method [36] or Quasi-Newton methods such as L-BFGS-B [37]. These methods often require many iterations to obtain a moderately accurate solution and thus

may be suitable for an inexact MM scheme where (14) is solved only approximately. However, such an inexact MM scheme typically slows down the rate of convergence of the iteration (14).

It is worth to note that $g(x)$ in (14) is a matrix fractional function, see [30, p. 76]. So each iteration (14) in fact involves solving a convex matrix fractional minimization problem [30], which is well-known to be equivalent to a semidefinite programming (SDP) problem

$$\begin{aligned} &\underset{z,x}{\text{minimize}} \quad z - \nabla h(x^{(k)})^T x, \quad z \in \mathbb{R}, x \in \mathbb{R}^p, \\ &\text{subject to} \quad \begin{bmatrix} z & Y^T \\ Y & \Sigma(x) \end{bmatrix} \succeq \mathbf{0}, \quad x \geq \mathbf{0}. \end{aligned}$$

The cost of solving this SDP is at least cubic in the number of hyper-parameters p as well as the number of observations n_o, and hence solving this SDP is too costly for all but small problems. Note that modeling packages such as CVX [38] commonly use such an SDP reformulation of a matrix fractional minimization problem.

From the definition of $\Sigma(x)$ in (16) and the constraint $x \geq \mathbf{0}$, we see that $\Sigma(x)$ is a sum of positive semidefinite terms. This implies that the matrix fractional minimization problem (14) can be cast as a conic quadratic optimization problem (see e.g. [39] and [40]). In particular, each iteration (14) amounts to solving the following conic optimization problem with $p+1$ rotated quadratic cone constraints (see [39, p. 202]), i.e.,

$$\begin{aligned} &\underset{z,x,v,w}{\text{minimize}} \; 2(\mathbf{1}^T z) - \nabla h(x^{(k)})^T x \\ &\text{subject to} \; \|w_i\|_2^2 \leq 2x_i z_i, \; i = 1, \ldots, p, \; \|v\|_2^2 \leq 2 \; z_{p+1} \\ &\qquad Y = \sum_{i=1}^{p} B_i w_i + \sigma v, \; x \geq \mathbf{0}, \; z \geq \mathbf{0} \end{aligned} \tag{17}$$

where $x \in \mathbb{R}^p$, $z = \begin{bmatrix} z_1 & \cdots & z_{p+1} \end{bmatrix}^T \in \mathbb{R}^{p+1}$, $v \in \mathbb{R}^{n_o}$, and $w_i \in \mathbb{R}^{n_i}$ for $i = 1, \ldots, p$. The problem (17), which is equivalent to a second-order cone program, can be solved efficiently and accurately using an interior-point method. The computational cost depends on the implementation. If the rotated quadratic cone constraints are handled carefully, the computational cost of a single interior-point iteration is $O({n_o}^2 \max(n_o, \sum_{i=1}^{p} n_i))$ and in particular linear in p if all n_is are equal; see e.g. [41]. We have implemented such a method for solving (17) in CVXOPT [42], a Python extension for convex optimization. Our implementation is based on the cone LP solver in CVXOPT, and uses a custom solver for the so-called KKT system that defines the search direction at each interior-point iteration, see [43]. The implementation details cannot be included here due to space limitations. The problem (17) can also be solved efficiently using, e.g., the commercial solver MOSEK.

Monte Carlo simulations in Section V show that, the proposed MM Algorithm 3.1 and implementation requires on average 12 iterations of (14) to obtain a high accuracy locally optimal solution of (13). Moreover, the locally optimal solutions lead to good performance, regardless of the initialization, which is a practical advantage over ML/PEM and KRM with nonlinearly parameterized kernels where the initialization is critical and tricky.

IV. Structure detection with multiple kernel-based regularization

Structure detection problems are in essence model structure selection problems in parameter space, e.g. [19], [20]. As will be seen in Sections IV-B and IV-C, using MKRM, the structure detection problems in [19], [20] are converted to problems of finding a suitable sparse pattern (*the number and the location of zeros*) of the hyper-parameter x, which can be seen as model structure selection problems in the hyper-parameter space. For convenience, we start the discussion from the problem of finding a suitable sparse pattern of x.

A. Finding a suitable sparse pattern of x

Since $x \in \mathbb{R}^p$, there are in total 2^p sparse patterns of x, denoted by $x^{[i]} \in \mathbb{R}^p$, and accordingly 2^p model structures denoted by $\mathscr{M}_i$, $i = 1, \cdots, 2^p$. Here $x^{[i]}$ should be understood as follows: some of its elements are locked to zero and the others are free variables. For example, for $p = 2$, there are four sparse patterns: $[0 \quad 0]^T$, $[x_1 \quad 0]^T$, $[0 \quad x_2]^T$ and $[x_1 \quad x_2]^T$.

In Bayesian framework, model structure selection problems are typically tackled by using the evidence maximization method (EMM), see e.g., [10]. Since all fixed kernels in (8) are instances of existing single kernels (7) that are independent of the data, it is natural to assume that the "subjective priors" $p(\mathscr{M}_i), i = 1, \cdots, 2^p$, are equal. In this case, $\mathscr{M}_i, i = 1, \cdots, 2^p$, are ranked by evaluating the evidence of $\mathscr{M}_i$, defined as $p(Y|\mathscr{M}_i) = \int p(Y|\mathscr{M}_i, x^{[i]}, \sigma^2) p(x^{[i]}|\mathscr{M}_i) p(\sigma^2|\mathscr{M}_i) dx^{[i]} d\sigma^2$, where $p(x^{[i]}|\mathscr{M}_i)$ and $p(\sigma^2|\mathscr{M}_i)$ are independent hyper-priors. EMM selects the model structure or equivalently the sparse pattern of x with the largest evidence as the best one, and moreover, has *the ability on the average to identify the true model structure*, see [10, p. 441]. There are however two practical difficulties for EMM. One is that the integral in $p(Y|\mathscr{M}_i)$ often has no closed-form solution. The other is that $p(Y|\mathscr{M}_i)$ may need to be computed for a large number of times. EMM is thus in general expensive to implement and only applicable for small scale problems in practice.

1) An efficient approximation of EMM: An approximation of EMM, which avoids the introduction of hyper-priors for $x^{[i]}$ and σ^2, was suggested in [44, p. 778]. For $i = 1, \cdots, 2^p$, let $\hat{x}^{[i]}, \hat{\sigma}_i^2 = \arg\max_{x^{[i]}, \sigma^2} p(Y|\mathscr{M}_i, x^{[i]}, \sigma^2) = \arg\min_{x^{[i]}, \sigma^2} f(x^{[i]}, \sigma^2)$, where $f(x^{[i]}, \sigma^2)$ is the objective function of (13) with x replaced by $x^{[i]}$. Then

$$\begin{aligned} \log p(Y|\mathscr{M}_i)/p(Y|\mathscr{M}_j) \approx \log p(Y|\mathscr{M}_i, \hat{x}^{[i]}, \hat{\sigma}_i^2) \\ - \log p(Y|\mathscr{M}_j, \hat{x}^{[j]}, \hat{\sigma}_j^2) - \frac{1}{2}(d_i - d_j)\log(n_o) \end{aligned} \tag{18}$$

where d_i and d_j are the numbers of nonzero elements of $[(\hat{x}^{[i]})^T \; \hat{\sigma}_i^2]^T$ and $[(\hat{x}^{[j]})^T \; \hat{\sigma}_j^2]^T$, respectively. Interestingly, minus twice of (18) is actually the Bayesian information criterion (BIC), see [44]. While (18) is more convenient to compute, the required computation in handling a model structure selection problem with 2^p model structures is still combinatorial. This difficulty can be overcome by noting the feature of the negative log marginal likelihood minimization problem (13):

Theorem 4.1: Consider (13). There exists a σ_{max}^2 such that for $\sigma^2 > \sigma_{max}^2$, the optimal solution of (13) is unique and

exactly zero. In particular, for the case where $B_iB_i^T$, $i=1,\cdots,p$, are nonsingular, let $\delta_i=\inf\{s|B_i^T(sI_{n_o}-YY^T)B_i\succ\mathbf{0}\}$, $i=1,\cdots,p$, and assume without loss of generality $+\infty=\delta_0>\delta_1\geq\delta_2\geq\cdots\geq\delta_p$. Then for each $i=0,\cdots,p-1$, if $\sigma^2\in(\delta_{i+1},\delta_i]$, every locally optimal solution of (13) contains at least $p-i$ zeros.

Proof: Denote the objective function of (13) by $f(x)$. Then for each $i=1,\cdots,p$, $\nabla_{x_i}f(x)=\mathrm{Tr}\left\{B_i^T\Sigma(x)^{-1}(\Sigma(x)-YY^T)\Sigma(x)^{-1}B_i\right\}$. Note that x is a stationary point of (13) if for each $i=1,\cdots,p$, either $\nabla_{x_i}f(x)=0$ for $x_i\geq 0$ or $\nabla_{x_i}f(x)>0$ for $x_i=0$. For convenience, define $\sigma^2_{max}=\inf\{s|sI_{n_o}-YY^T\succ\mathbf{0}\}$. Then for any $\sigma^2>\sigma^2_{max}$, $\nabla_{x_i}f(x)>0$ for each $i=1,\cdots,p$ and any $x\geq\mathbf{0}$, and thus (13) has a unique optimal solution and is exactly zero. What's more, if $B_iB_i^T$, $i=1,\cdots,p$, are nonsingular,

$$\nabla_{x_i}f(x)=\mathrm{Tr}\{B_i^T\Sigma(x)^{-1}(B_iB_i^T)^{-1}B_iB_i^T(\sum_{k=1}^{p}x_kB_kB_k^T+\sigma^2I_{n_o}-YY^T)B_iB_i^T(B_iB_i^T)^{-1}\Sigma(x)^{-1}B_i\}.$$

For each $i=0,\cdots,p-1$, if $\sigma^2\in(\delta_{i+1},\delta_i]$, $\nabla_{x_j}f(x)>0$ for $j=i+1,\cdots,p$ and $x\geq\mathbf{0}$, which implies all locally optimal solutions have $x_j=0$, $j=i+1,\cdots,p$. We thus complete the proof.

Remark 4.1: It should be noted that no constraint is imposed on n_o and p in Theorem 4.1. If $n_o<p$, as shown in [27, Thm. 1], for any $\sigma^2\geq 0$, every locally optimal solution of (13) is achieved at a sparse solution with at most n_o nonzeros. The proof of this claim is a straightforward extension of that of [12, Thm. 2] to the general multiple kernel (8). Theorem 4.1 and this claim indicate that the negative log marginal likelihood minimization problem (13) has an inherent mechanism of favoring sparse hyper-parameters.

From Theorem 4.1 and Remark 4.1, we have the following efficient way to find a suitable sparse pattern of x, which is referred to as MKRM-BIC below. First solve

$$\hat{x}^{[0]},\hat{\sigma}_0^2=\arg\min_{x,\sigma^2}f(x,\sigma^2). \tag{19}$$

Then, set $k=0$ and go to the next iterative steps:

a) Determine $x^{[k+1]}$ as follows: $x^{[k+1]}$ is similar to $x^{[k]}$ with the only difference that $x^{[k+1]}$ has one more zero that corresponds to the smallest nonzero element of $\hat{x}^{[k]}$;
b) Invoking (18), if $\log p(Y|\mathscr{M}_k)/p(Y|\mathscr{M}_{k+1})>0$, stop and select $x^{[k]}$ as the best sparse pattern; if otherwise, set $k=k+1$ and go to step a).

2) Two heuristic methods: Noticing the form of (18), we consider two heuristic methods that are also based on the marginal likelihood maximization. The idea of the first one is to solve (19) and select the sparse pattern $x^{[0]}$ as a reference, and then trim $x^{[0]}$ by removing the small nonzeros that has little influence on the marginal likelihood. This heuristic method is referred to as MKRM-H1 and is detailed as follows. First, solve (19) and let $o_0=f(\hat{x}^{[0]},\hat{\sigma}_0^2)$. Then, set the threshold $o_h>0$, $k=0$ and go to the next iterative steps:

a) Determine $x^{[k+1]}$ as follows: $x^{[k+1]}$ is similar to $x^{[k]}$ with the only difference that $x^{[k+1]}$ has one more zero that corresponds to the smallest nonzero element of $\hat{x}^{[k]}$;
b) Solve $\hat{x}^{[k+1]},\hat{\sigma}_{k+1}^2=\arg\min_{x^{[k+1]},\sigma^2}f(x^{[k+1]},\sigma^2)$ and let $o_{k+1}=f(\hat{x}^{[k+1]},\hat{\sigma}_{k+1}^2)$. Check if $|(o_{k+1}-o_0)/o_0|>o_h$: if yes, stop and select $x^{[k]}$ as the best sparse pattern; if otherwise, set $k=k+1$ and go to step a).

Here, the threshold o_h is a tuning parameter and can be tuned, e.g., by cross validation.

Both MKRM-BIC and MKRM-H1 rely on $\hat{\sigma}_0^2$, the estimate of σ^2 by maximizing the marginal likelihood (19), which can however be very inaccurate. It can even happen that $\hat{\sigma}_0^2=0$ because there can exist nonzero x such that $\sum_{i=1}^p x_iB_iB_i^T$ has identical contribution as $\sigma^2I_{n_o}$ on $\Sigma(x)$ in (16), see [45, Section 3.C] for related discussions on basis selection problems. For the segmentation problem in Section IV-C, $\hat{\sigma}_0^2$ is often much smaller than necessary due to the use of the over-parameterized model (23). In this case, the sparse pattern $x^{[0]}$ in (19) is very inaccurate, and hence MKRM-BIC and MKRM-H1 should not be used. As suggested in [45, Section 3.C] for basis selection problems, an alternative way is to tune the sparse pattern of x by tuning σ^2 and solving (13) accordingly, which is possible by noting Theorem 4.1 and Remark 4.1. As for which sparse pattern is more suitable, one can use application dependent heuristic [45], cross validation [19], and EMM [10] if possible. This heuristic method is referred to as MKRM-H2 below.

B. Sparse dynamic network identification [19]

1) Problem statement and formulation: Consider a multiple-input-single-output (MISO) linear stable system

$$y(t)=\sum_{j=1}^{r}G_j(q)u_j(t)+v(t), \tag{20}$$

where $y(t)$ and q are defined as in (1), $u_j(t)$ is the input for the jth subsystem $G_j(q)$, and $v(t)$ is a white noise with mean zero and variance σ^2, independent of $u_j(t),j=1,\cdots,r$. The assumption is that there exists an index set $\mathscr{I}\subset\{1,\cdots,r\}$ such that the inputs $u_j(t)$ with $j\in\mathscr{I}$ do not influence $y(t)$, i.e., the corresponding $G_j(q)$ are zero. The goal is to estimate the index set $\mathscr{I}$ with a given data record $\{u_1(t),\cdots,u_r(t),y(t)\}_{t=1}^N$.

To tackle the problem, subsystems $G_j(q)$, $j=1,\cdots,r$, are modeled as FIR models

$$G_j(q,\theta_j)=\sum_{k=1}^{n}g_{k,j}q^{-k},\theta_j=\begin{bmatrix}g_{1,j} & g_{2,j} & \cdots & g_{n,j}\end{bmatrix}^T. \tag{21}$$

Like (4), $Y_N=\sum_{j=1}^r\Phi_{N,j}^T\theta_j+V_N$ where Y_N,V_N are defined in (4), and $\Phi_{N,j}$ is defined similarly as Φ_N by replacing u with u_j. Then the problem is tackled by using the following MKRM:

$$\hat{\theta}_1,\cdots,\hat{\theta}_r=\arg\min_{\theta_1,\cdots,\theta_r}\|Y_N-\sum_{j=1}^{r}\Phi_{N,j}^T\theta_j\|_2^2+\sigma^2\sum_{j=1}^{r}\theta_j^TP(\alpha_j)^{-1}\theta_j \tag{22}$$

where $P(\alpha_j)=\sum_{i=1}^m c_{i,j}P_i$, $\alpha_j=\begin{bmatrix}c_{1,j},\cdots,c_{m,j}\end{bmatrix}^T$, $P_i\succeq\mathbf{0}$, $i=1,\cdots,m$, are fixed kernels and $c_{i,j}\geq 0$, $i=1,\cdots,m$, are the hyper-parameters associated with θ_j. If $\alpha_j=\mathbf{0}$ for some $j=1,\cdots,r$, $P(\alpha_j)=\mathbf{0}$ and thus $\hat{\theta}_j=\mathbf{0}$, which indicates $G_j(q,\hat{\theta}_j)=0$. In this way, the problem is converted to a problem of finding a suitable sparse pattern of hyper-parameters.

2) Finding a suitable sparse pattern of hyper-parameters: The regularization term $\sum_{j=1}^{r}\theta_j^T P(\alpha_j)^{-1}\theta_j$ in (22) is first embedded in Bayesian framework. Assume $v(t)\sim\mathscr{N}(0,\sigma^2)$, $\theta_j\sim\mathscr{N}(\mathbf{0},P(\alpha_j))$ and moreover, they are independent from each other. It can be shown that the MAP estimation problem $\arg\max_{\theta_1,\cdots,\theta_r} p(\theta_1,\cdots,\theta_r|Y_N)$ is equivalent to (22). Noting the factorization of $P_i=L_iL_i^T, i=1,\cdots,m$, the marginal likelihood maximization problem $\text{maximize}_{\alpha_1,\cdots,\alpha_r} p(Y_N|\alpha_1,\cdots,\alpha_r)$ can be put into the form (13) with $p=mr$, $n_o=N-n$, $Y=Y_N$, $x=\begin{bmatrix}\alpha_1^T & \cdots & \alpha_r^T\end{bmatrix}^T$ and $B_{(j-1)m+i}=\Phi_{N,j}^T L_i,\ i=1,\cdots,m,\ j=1,\cdots,r$. This problem is handled by using MKRM-BIC and MKRM-H1 in Section IV-A.

Remark 4.2: In theory, single kernels (7) can be used to tackle the sparse dynamic network identification problem. In practice, they however cannot be applied due to the difficulty of the solution of the associated marginal likelihood maximization problem. To overcome this difficulty, the problem was handled in [19] using a variant of KRM in [4], [5], where SS kernels (7a) for different subsystems are assumed to have the same β in (7a), i.e., $\theta_j\sim\mathscr{N}(\mathbf{0},c_jP^{ss}(\beta))$, $j=1,\cdots,r$. Moreover, exponential hyper-priors on c_j, i.e., $p(c_j)=\gamma\exp(-\gamma c_j)$ with $\gamma\geq 0$ are imposed to enhance the sparsity of $c_1,\cdots,c_r$, which is achieved by solving the MAP problem $\text{maximize}_{c_1,\cdots,c_r,\beta} p(c_1,\cdots,c_r,\beta|Y_N)$ with a suitable γ, tuned by cross validation. The non-convex MAP problem has no special structure and is handled by using a Quasi-Newton algorithm. An important issue for the numerical algorithm is the availability of a good starting point, which is provided by a Bayesian forward selection algorithm. In contrast, employing the multiple kernel (8) and accordingly the MM Algorithm 3.1 and implementation greatly simplifies the solution but yields comparable performance as the method in [19].

C. Segmentation of linear systems [20]

1) Problem statement and formulation: Consider system (1). The assumption is that $G_0(q)$ changes its dynamics at certain time instants which are rare. The goal is to detect the changes with a given data record $\{(y(t),u(t))\}_{t=1}^N$.

Since $G_0(q)$ may change at any time instant, we associate with each time instant t an FIR model with parameter vector θ_t, that is,

$$y(t)=\phi(t)^T\theta_t+v(t),\qquad t=n+1,\cdots,N, \tag{23}$$

where $\phi(t)$ is defined in (3) and $\theta_t=\begin{bmatrix}g_{1,t} & g_{2,t} & \cdots & g_{n,t}\end{bmatrix}^T$. Define $\theta_n=\mathbf{0}$. Then the problem is tackled by using the following MKRM:

$$\begin{aligned}\hat\theta_{n+1},\cdots,\hat\theta_N=\underset{\theta_{n+1},\cdots,\theta_N}{\arg\min}&\sum_{t=n+1}^{N}(y_t-\phi(t)^T\theta_t)^2\\&+\sigma^2(\theta_t-\theta_{t-1})^TP(\alpha_t)^{-1}(\theta_t-\theta_{t-1}),\end{aligned} \tag{24}$$

where $P(\alpha_t)=\sum_{i=1}^m c_{i,t}P_i$, $\alpha_t=[c_{1,t},\cdots,c_{m,t}]^T$, P_i, $i=1,\cdots,P_m$, are fixed kernels in (8) and $c_{i,t}\geq 0$, $i=1,\cdots,m$, are hyper-parameters associated with θ_t. If $\alpha_t=\mathbf{0}$ for some $t=n+1,\cdots,N$, then $\theta_t=\theta_{t-1}$ and the corresponding term $\sigma^2(\theta_t-\theta_{t-1})^TP(\hat\alpha_t)^{-1}(\theta_t-\theta_{t-1})$ would disappear from (24). Therefore, $\alpha_t\neq\mathbf{0}$ for certain $t=n+1,\cdots,N$, is an indication that the dynamics of system (1) changes at time t. In this way, the segmentation problem is converted to a problem of finding a suitable sparse pattern of hyper-parameters.

2) Finding a suitable sparse pattern of hyper-parameters: The regularization term $\sum_{t=n+1}^{N}(\theta_t-\theta_{t-1})^TP(\alpha_t)^{-1}(\theta_t-\theta_{t-1})$ in (24) is first embedded in Bayesian framework. Assume $v(t)\sim\mathscr{N}(0,\sigma^2)$, independent of $\theta_t\sim\mathscr{N}(\theta_{t-1},P(\alpha_t))$, $t=n+1,\cdots,N$. Then it can be shown that the MAP estimation problem $\arg\max_{\theta_{n+1},\cdots,\theta_N} p(\theta_{n+1},\cdots,\theta_N|Y_N)$ is equivalent to (24). Moreover, the marginal likelihood $p(Y_N|\alpha_{n+1},\cdots,\alpha_N)=\mathscr{N}(\mathbf{0},K(\alpha_{n+1},\cdots,\alpha_N,\sigma^2))$, where for $t=n+1,\cdots,N-1$,

$$\begin{aligned}K(\alpha_t,\cdots,\alpha_N,\sigma^2)&=\begin{bmatrix}\sigma^2 & \mathbf{0}\\ \mathbf{0} & K(\alpha_{t+1},\cdots,\alpha_N,\sigma^2)\end{bmatrix}\\&+\begin{bmatrix}\phi(t)^T\\ \vdots\\ \phi(N)^T\end{bmatrix}P(\alpha_t)\begin{bmatrix}\phi(t) & \cdots & \phi(N)\end{bmatrix}\\K(\alpha_N,\sigma^2)&=\phi(N)^TP(\alpha_N)\phi(N)+\sigma^2.\end{aligned} \tag{25}$$

Noting (25) and the factorization of the fixed kernels $P_i=L_iL_i^T, i=1,\cdots,m$, the marginal likelihood maximization problem $\text{maximize}_{\alpha_{n+1},\cdots,\alpha_N} p(Y_N|\alpha_{n+1},\cdots,\alpha_N)$ can be put into the form of (13) with $p=m(N-n)$, $n_o=N-n$, $Y=Y_N$, $x=\begin{bmatrix}\alpha_{n+1}^T & \cdots & \alpha_N^T\end{bmatrix}^T$ and $B_{tm+i}=\begin{bmatrix}\mathbf{0}_{n\times t} & \phi(n+t+1) & \cdots & \phi(N)\end{bmatrix}^TL_i$, $t=0,\cdots,N-n-1, i=1,\cdots,m$. The problem cannot be handled by using either MKRM-BIC or MKRM-H1 in Section IV-A, because solving (19) often yields much smaller $\hat\sigma_0^2$ than necessary, and the sparse pattern $x^{[0]}$ is thus very inaccurate. Instead, the problem is handled by using MKRM-H2 in Section IV-A.

Remark 4.3: In our notations, the segmentation problem was formulated in [20] as

$$\hat\theta_{n+1},\cdots,\hat\theta_N=\underset{\theta_{n+1},\cdots,\theta_N}{\arg\min}\sum_{t=n+1}^{N}(y_t-\phi(t)^T\theta_t)^2+\gamma||\theta_t-\theta_{t-1}||_2,\gamma\geq 0$$

which can be seen as a variant of Group LASSO. In contrast, MKRM induces sparsity in the hyper-parameter space, which in turn results in sparsity in the parameter space.

V. Numerical illustrations

The proposed MKRM, MM Algorithm 3.1 and implementation are tested for the model estimation problem in Section II and the structure detection problems in Sections IV-B and IV-C. Before proceeding to the details, some common settings for all simulations are given.

Generic systems. Generic systems should be representatives of real-life systems in that the underlying system is not of low order but could allow good low order approximations. The generic system that will be tested is generated in the same way as detailed below. A SISO continuous-time system of 30th order is first generated using the command `m=rss(30)` in MATLAB. The continuous-time system `m` is then sampled at 3 times of its bandwidth to yield the corresponding discrete-time system `md` using the following commands in MATLAB: `bw=bandwidth(m); f = bw*3*2*pi; md=c2d(m,1/f,'zoh')`. If all poles of `md` are within

the circle with center at the origin and radius 0.95, set the feedthrough matrix of `md` to 0 and save it as one generic system.

Unknown initial conditions. Using $y(t), t = 1, \cdots, n$ requires the unknown $u(1-n), \cdots, u(0)$, which can be handled in different ways. For convenience, $y(t), t = 1, \cdots, n$, will not be used for KRM for model estimation and MKRM for both model estimation and structure detection.

Implementation. All marginal likelihood maximization problems are tackled with the MM Algorithm 3.1 and our custom interior-point method in Python using CVXOPT, see Section III-C. For MKRM, all initializations are randomly generated in MATLAB as $x^{(0)} =$ `abs(5*randn(p,1))` and $\sigma^{2(0)} =$ `abs(5*randn(1))`.

A. Model estimation

1) Data-bank of test systems and data sets: The data-bank consists of four collections of 1000 generic systems and data sets: D1, D2, D3 and D4. For each generic system in D1, the associated data set contains 210 data points and is generated as follows: simulate the generic system with an input which is white Gaussian noise with unit variance, and an output additive white Gaussian noise whose variance is one tenth of the variance of the noise-free output. D2 is generated similarly as D1 with the only difference that the output additive white Gaussian noise has the same variance as the noise-free output. D3 and D4 are generated similarly as D1 and D2, respectively, with the difference that a band-limited random Gaussian signal is used to simulate the generic system and moreover, each data set contains 500 data points. The band-limited random Gaussian signal is generated using the command `idinput` in [46] with band $[0, 0.8]$, where 0 and 0.8 are the lower and upper limits of the pass band, expressed as fractions of the Nyquist frequency.

2) Simulation setup and results: The order of the FIR model (2) is set to 100 and two multiple kernels (8) are generated based on the following collections of fixed kernels:

- 54 DC kernels (7c). They are obtained by evaluating (7c) on the grid with $c = 1$, $\lambda = 0.1 : 0.1 : 0.9$, and $\rho = -0.95, -0.65, -0.35, 0.35, 0.65, 0.95$.
- 21 TC kernels (7b) and 8 SS kernel (7a). The 21 TC kernels are obtained by evaluating (7b) on the grid with $c = 1$ and $\lambda = 0.1 : 0.05 : 0.75, 0.81 : 0.02 : 0.93$. The 8 SS kernels are obtained by evaluating (7a) on the grid with $c = 1$ and $\lambda = 0.8 : 0.02 : 0.94$.

In fact, P_i can also be constructed based on an available model estimate $G(q)$. This idea is motivated by the form of the optimal kernel that minimizes the MSE matrix $\mathscr{E}[(\hat{\theta}_N^R - \theta_0)(\hat{\theta}_N^R - \theta_0)^T]$. According to [6, Thm. 1], $\mathscr{E}[(\hat{\theta}_N^R - \theta_0)(\hat{\theta}_N^R - \theta_0)^T]$ is minimized at $P^{opt} = \theta_0\theta_0^T$. So it is natural to construct $P_i = \hat{\theta}(G(q))(\hat{\theta}(G(q)))^T$, where $\hat{\theta}(G(q))$ is the column vector containing the first n impulse response coefficients of $G(q)$. Note that $G(q)$ can be either data-driven or data-free. For example, it can be an output error (OE) model $G(q, \hat{\theta}_N^{oe})$ with a suitable order estimated based on $\{u(t), y(t)\}_{t=1}^N$, see [1]. For illustration, we consider the third multiple kernel (8) with the collection of fixed kernels:

- 6 kernels in the form of $\hat{\theta}(G(q, \hat{\theta}_N^{oe}))(\hat{\theta}(G(q, \hat{\theta}_N^{oe})))^T$, where $G(q, \hat{\theta}_N^{oe})$ are OE model estimates of order 2 to 7 using the `oe` command in [46]. Note that for this kind of fixed kernels, the factorization $L_i = \hat{\theta}(G(q, \hat{\theta}_N^{oe}))$ and $n_i = 1$ in the derivation of (13).

Remark 5.1: Assume $P_i = \hat{\theta}(G_i(q))(\hat{\theta}(G_i(q)))^T$, $i = 1, \cdots, m$, where $G_i(q)$, $i = 1, \ldots, m$ are some available model estimates. Solving (11) yields $\hat{\alpha} = \left[\hat{c}_1, \cdots, \hat{c}_m\right]^T$, $P(\hat{\alpha}) = \sum_{i=1}^m \hat{c}_i \hat{\theta}(G_i(q))(\hat{\theta}(G_i(q)))^T$, and $\hat{\theta}_N^R = \sum_{i=1}^m a_i \hat{c}_i \hat{\theta}(G_i(q))$, where $a_i = (\hat{\theta}(G_i(q)))^T(\Phi_N \Phi_N^T P(\hat{\alpha}) + \sigma^2 I_n)^{-1}\Phi_N Y_N$, $i = 1, \cdots, m$. Since a_i is a scalar, $\hat{\theta}_N^R$ is a weighted average over the impulse responses of the model estimates $G_i(q)$, $i = 1, \cdots, m$. That means for this multiple kernel, MKRM is closely related with the composite modeling in [47]. It is also interesting to note that only some of $G_i(q)$ actually contribute to $\hat{\theta}_N^R$ since $\hat{\alpha}$ is often sparse.

The three multiple kernels are denoted below by "DC-M", "TCSS-M" and "OE(2:7)-M", respectively. The noise variance σ^2 is estimated together with α by maximizing the marginal likelihood and the impulse response estimate $\hat{\theta}_N^R$ is computed according to (5b). The proposed approach is compared with the KRM [4]–[6] with kernels (7) where the implementation in [48] is used. It is also compared with the ML/PEM (the `oe` command in [46] is used) equipped with both AIC (Akaike's information criterion) and cross validation (CV) to select the best model order testing orders $1 : 1 : 30$. To evaluate various approaches, the impulse response estimates $\hat{g}_k, k = 1, \cdots, n$, are compared to the true ones by the measure

$$W = 100\left(1 - \left[\frac{\sum_{k=1}^{100}|g_k^0 - \hat{g}_k|^2}{\sum_{k=1}^{100}|g_k^0 - \bar{g}^0|^2}\right]^{1/2}\right), \quad \bar{g}^0 = \frac{1}{100}\sum_{k=1}^{100} g_k^0. \tag{26}$$

where W corresponds to the "fit" in the `compare` command in [46]. The results are shown in the following table where "AF" denotes the average fit (26) and "NO" denotes the number of outliers below zero for the associated data collection:

AF|NO	PEM-AIC	PEM-CV	TC	SS	DC	OE(2:7)-M	DC-M	TCSS-M
D1	81.9|0	83.8|9	81.5|0	82.1|0	82.1|0	86.6|0	84.4|0	84.4|0
D2	43.6|67	61.8|16	55.9|25	56.1|6	54.3|24	61.1|0	63.2|0	63.7|0
D3	-1334.2|378	46.0|95	87.6|0	87.7|0	87.9|0	80.2|17	87.6|0	88.8|0
D4	-3904.9|470	2.8|198	74.6|0	73.4|0	74.9|0	39.8|140	74.8|1	76.7|0

It is interesting to study the distribution of the fits over individual data collections, which are shown by box-plots in Fig. 2. As can be seen from the table and Fig. 2, for all four data collections there is always one multiple kernel, for which the MKRM outperforms the other approaches. Moreover, MKRM with TCSS-M gives the overall best performance.

Remark 5.2: It should be noted that for PEM-AIC and PEM-CV, the `oe` command in MATLAB uses all data points $\{u(t), y(t)\}_{t=1}^N$. If all data points are used for KRM and MKRM and the unknown $u(1-n), \cdots, u(0)$ are set to zero, the performance of KRM and MKRM can be further improved. For example, the average fit for MKRM with DC-M and TCSS-M increases from 84.4 to 87.1 and 86.9, respectively. It is also interesting to note that PEM works much worse for D3 and D4 than KRM and MKRM. The reason is that with band limited input, there is not full information in the data about the impulse response, and kernel-based regularization methods benefit from the smoothness assumption implicitly present in the regularization.

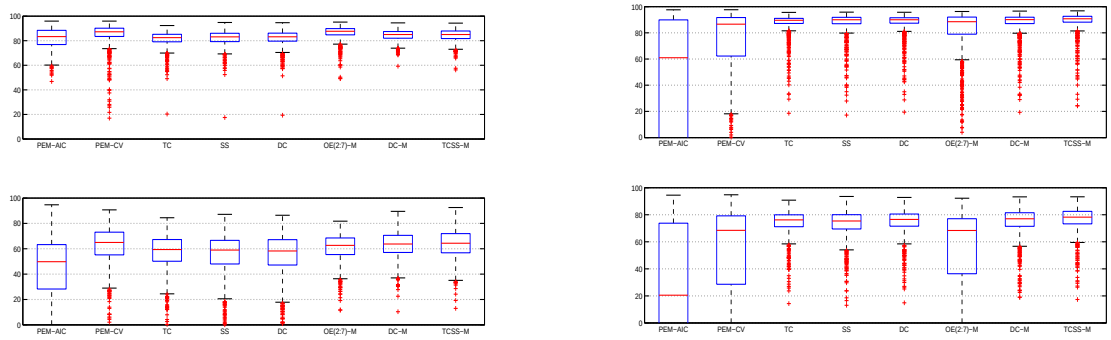

Fig. 2. Box-plots of the 1000 fits: Left plot for D1 (top) and D2 (bottom); Right plot for D3 (top) and D4 (bottom).

Remark 5.3: When using TC-M and TCSS-M, as the grid density increases from small to large, the performance tends to increase but it will remain virtually the same after certain point even if a more dense grid is used. In fact, overfitting is avoided thanks to the use of the marginal likelihood maximization for hyper-parameter estimation. What's more, even for model estimation, TC-M and TCSS-M often have sparse hyper-parameters. For illustration, the average number of "alive" fixed kernels in TCSS-M for the four data collections are 3.92, 2.95, 5.20, and 3.58, respectively, where the P_i in (8) with $\hat{c}_i >$1e-5 is identified as alive.

B. Sparse dynamic network identification

1) Data-bank of test systems and data sets: The data-bank consists of two collections of 500 data sets: D5 and D6. Each data set in D5 contains 600 data points and is generated as follows. First, 10 generic systems are generated. The command `zInd = unique(sort(randi(10,1,10),'ascend')); Ind = ones(10,1);Ind(zInd) = 0;` in MATLAB is then used to generate `Ind`. Each element of `Ind` describes if the corresponding input or generic system has influence on the overall output: "1" means true and "0" means otherwise. Those systems which have influence on the overall output are simulated individually with an input which is white Gaussian noise with unit variance. Then the individual simulated outputs are summed and the sum is regarded as the overall noise free output. Further the noise free output is perturbed by an additive white Gaussian noise whose variance is one tenth of the variance of the noise free output. D6 is generated similarly as D5 with the only difference that the output additive white Gaussian noise has the same variance as the noise-free output.

2) Simulation setup and results: The order of all FIR models (21) is set to 100 and the multiple kernel (8) is generated based on 6 TC kernels (7b), which are obtained by evaluating (7b) on the grid with $c = 1$ and $\lambda = 0.82 : 0.02 : 0.92$. As shown in Section IV-B, the problem is converted to a problem of finding a suitable sparse pattern of hyper-parameters and handled by using MKRM-BIC and MKRM-H1 in Section IV-A. In particular for MKRM-H1, o_h is set to 8e-3 for D5 and 3e-3 for D6. Here, MKRM is compared with the stable spline exponential hyper-prior (SSEH) approach [19] and the group LAR [9], which are implemented as described in [19]. The percentage of whether the inputs have influence on the output or not is correctly identified is summarized in the following table, which shows that, MKRM works comparably as SSEH and they all behave better than Group LAR.

Data	Group LAR	SSEH	MKRM-BIC	MKRM-H1
D5	83.5%	98.0%	98.3%	99.0%
D6	81.7%	94.1%	90.6%	93.9%

Remark 5.4: For MKRM-H1, the threshold o_h is tuned on a small number of data sets by cross validation. It is reasonable to have smaller o_h for D6 because each data set has larger noise and leads to smaller estimate of the hyper-parameters, which in turn has less influence on the marginal likelihood.

C. Segmentation of linear systems

1) Test data set: First, two generic systems `M1` and `M2` and a white Gaussian noise input `u=randn(500,1)` in MATLAB are generated. The system `M1` is simulated with the input `u` and the simulated output is denoted by `ynf1`. At the 301st time instant, the other system `M2` is switched on and is simulated as follows: `ynf2=ynf1(301)+sim(M2,u(301:500))` where `ynf2` denotes the simulated output. Then set `ynf=[ynf1;ynf2]` as the noise free output. The measurement output is then collected by disturbing the noise free output with an output additive white Gaussian noise whose variance is one tenth of the variance of the noise free output. In this way we get the test data set which contains 500 data points and has the change occurring at $t^\star = 301$. The impulse response of the two generic systems together with their difference and the profile of the measurement output are shown in Fig. 3.

2) Simulation setup and results: The order of FIR models (23) at each time instant t is set to 100. The multiple kernel (8) is generated based on 3 TC kernels (7b), which are obtained by evaluating (7b) on the grid with $c = 1$ and $\lambda = 0.8 : 0.06 : 0.92$. As shown in Section IV-C, the problem is converted to a problem of finding a suitable sparse pattern of hyper-parameters and handled by using MKRM-H2 in Section IV-A. Since there is only one change, we can simply tune the sparse pattern of hyper-parameters by tuning σ^2 and solving (13) such that there is only one $\hat{\alpha}_t = [\hat{c}_{1,t}, \hat{c}_{2,t}, \hat{c}_{3,t}]^T \neq \mathbf{0}$ among $t = 101, \cdots, 500$. That means that the system changes its dynamics at that time instant. It turns out that $\sigma^2 = 6$ is a suitable choice. From the profile of $\sum_{i=1}^{3} \hat{c}_{i,t}$, $t = 101, \cdots, 500$, in Fig. 3, obtained by solving (13) with $\sigma^2 = 6$, we see clearly the system changes its dynamics around $t = 301$.

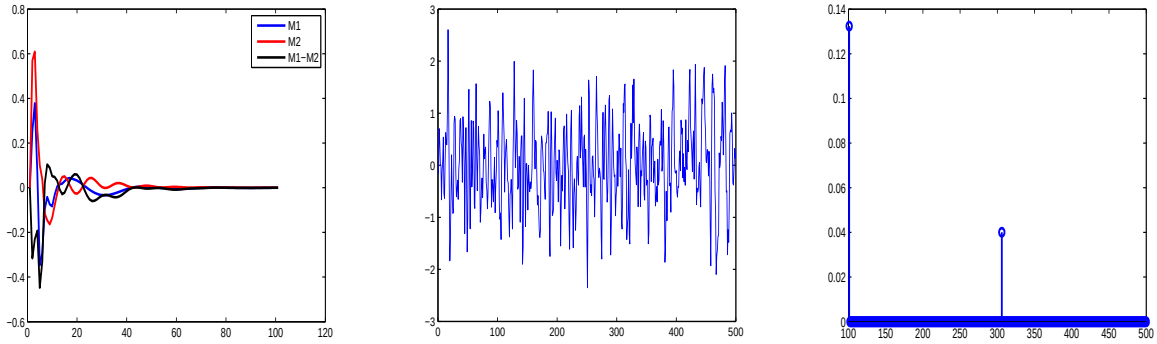

Fig. 3. The impulse response of the two generic systems and their difference (left), the measurement output (middle) and the profile of $\sum_{i=1}^{3}\hat{c}_{i,t}$, $t=101,\cdots,500$, obtained by solving (13) with $\sigma^2=6$ (right).

VI. Conclusions

While kernel techniques have been used for quite some time in linear regression model estimation problems in statistics and machine learning, they have only recently been introduced in the system identification literature. This has led to several contributions on how to choose suitable kernels for identification applications. In this paper we have discussed the use of multiple kernels and pointed to three distinct advantages with such a choice.

Firstly, that they can handle estimation of models with complicated dynamics, e.g., with widely spread time constants, better than well-tuned single kernels.

Secondly, that estimation of their weights by maximizing the marginal likelihood has an inherent feature of favoring sparse optimal weights. This method thus has an interesting potential for structure detection problems, such as finding the most important links in networked systems, and segmentation of time-varying systems.

Thirdly, the marginal likelihood maximization problem is a difference of convex programming problem, whose locally optimal solutions can be found efficiently using sequential convex optimization techniques. In particular, each subproblem can be solved efficiently using an interior-point method where the cost of a single interior-point iteration grows linearly in the number of fixed kernels. Monte Carlo simulations show that the locally optimal solutions lead to good performance, regardless of the initialization, which is a practical advantage over the maximum likelihood/prediction error method and the kernel-based regularization method with nonlinearly parameterized kernels where the initialization is critical and tricky.

A key issue to use multiple kernels is how to design suitable fixed kernels. A simple but effective way is to use the state of art single kernels SS, TC and DC, see (7): introduce a grid (could be uniform if there is no other prior knowledge about the unknown system) over the compact domain of β and generate fixed kernels on the points of the grid. As the grid density (the number of fixed kernels) increases from small to large, the performance tends to increase but it will remain virtually the same for TC and SS kernels after certain point even if a more dense grid is used. In contrast with DC kernel, both TC and SS kernels have dim $\beta=1$, which becomes advantageous in the design of fixed kernels in a computational perspective. Moreover, TC and SS kernels enjoy some interesting maximum entropy properties [17]. In some sense, they represent the least committing Bayesian priors when regularity and stability is the only information about the unknown system. Hence, in the design of fixed kernels for system identification, the combination of TC and SS, adopted by TCSS-M, appears a natural and efficient choice. Instead of SS, TC and DC kernels, different kernels could be however used when more information on the unknown system is available. For instance, when identifying relaxation systems, it could be advantageous to resort to kernels whose sections are completely monotonic (one example is the exponential kernel in [28, Eq. (4.2)]).

Motivated by the form of the optimal kernel in the sense of minimizing the mean square error, another way to design fixed kernels is to use rank-1 kernels $\hat{\theta}(G(q))(\hat{\theta}(G(q)))^T$, see Section V-A, where $G(q)$ can be either data-driven or data free. Design of data-driven rank-1 kernels is more involved and interested readers are referred to [49] for initial discussions. An interesting research topic is how to construct multiple data-free rank-1 kernels in an efficient way and enrich the multiple SS, TC and DC kernels with the data-free rank-1 kernels. In fact, this topic is closely related with the compressive sensing and basis selection [12].

References

[1] L. Ljung, *System Identification - Theory for the User*. Upper Saddle River, N.J.: Prentice-Hall, 2nd ed., 1999.
[2] T. Söderström and P. Stoica, *System Identification*. London: Prentice-Hall Int., 1989.
[3] R. Pintelon and J. Schoukens, *System Identification: A frequency domain approach*. Wiley-IEEE Press, 2012.
[4] G. Pillonetto and G. D. Nicolao, "A new kernel-based approach for linear system identification," *Automatica*, vol. 46, no. 1, pp. 81–93, 2010.
[5] G. Pillonetto, A. Chiuso, and G. D. Nicolao, "Prediction error identification of linear systems: a nonparametric Gaussian regression approach," *Automatica*, vol. 47, no. 2, pp. 291–305, 2011.
[6] 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.
[7] L. Ljung, H. Hjalmarsson, and H. Ohlsson, "Four encounters with system identification," *European Journal of Control*, vol. 17, pp. 449–471, 2011.
[8] 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.
[9] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[10] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.
[11] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, pp. 211–244, 2001.
[12] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153 – 2164, 2004.
[13] D. Wipf and S. Nagarajan, "Iterative reweighted and methods for finding sparse solutions," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 317 –329, 2010.
[14] A. Aravkin, J. Burke, A. Chiuso, and G. Pillonetto, "Convex vs non-convex estimators for regression and sparse estimation: the mean squared error properties of ARD and GLasso," *Journal of Machine Learning Research*, 2014.
[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.
[16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer, 2nd ed., 2009.
[17] G. Pillonetto and G. D. Nicolao, "Kernel selection in linear system identification. Part I: A Gaussian process perspective," in *Proc. 50th IEEE Conference on Decision and Control and European Control Conference*, (Orlando, Florida), pp. 4318–4325, 2011.
[18] T. Chen, H. Ohlsson, and L. Ljung, "Kernel selection in linear system identification. Part II: A classical perspective," in *Proc. 50th IEEE Conference on Decision and Control and European Control Conference*, (Orlando, Florida), pp. 4326–4331, 2011.
[19] A. Chiuso and G. Pillonetto, "A Bayesian approach to sparse dynamic network identification," *Automatica*, vol. 48, no. 8, pp. 1553 – 1565, 2012.
[20] H. Ohlsson, L. Ljung, and S. Boyd, "Segmentation of ARX-models using sum-of-norms regularization," *Automatica*, vol. 46, pp. 1107–1111, Apr. 2010.
[21] P. D. Tao and L. T. H. An, "Convex analysis aprroach to D. C. programming: Theory, Algorithms and Applications," *ACTA Mathematica Vietnamica*, vol. 22, pp. 289–355, 1997.
[22] R. Horst and N. V. Thoai, "DC programming: Overview," *Journal of Optimization Theory and Applications*, vol. 103, no. 1, pp. 1–43, 1999.
[23] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (CCCP)," *Advances in Neural Information Processing Systems*, vol. 2, pp. 1033–1040, 2002.
[24] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *American Statistician*, vol. 58, pp. 30–37, 2004.
[25] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, 2004.
[26] T. Evgeniou, C. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.
[27] D. Wipf and S. Nagarajan, "A unified Bayesian framework for MEG/EEG source imaging," *Neuroimage*, vol. 44, no. 3, pp. 947–966, 2009.
[28] F. Dinuzzo, "Kernels for linear time invariant system identification," *CoRR*, vol. abs/1203.4930, 2012.
[29] B. P. Carlin and T. A. Louis, *Bayes and Empirical Bayes methods for data analysis*. London: Chapman & Hall, 1996.
[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, England: Cambridge University Press, 2004.
[31] G. C. Goodwin, M. Gevers, and B. Ninness, "Quantifying the error in estimated transfer functions with application to model order selection," *IEEE Trans. Automatic Control*, vol. 37, no. 7, pp. 913–929, 1992.
[32] L. Ljung and B. Wahlberg, "Asymptotic properties of the least-squares method for estimating transfer functions and disturbance spectra," *Adv. Appl. Prob.*, vol. 24, pp. 412–440, 1992.
[33] B. Sriperumbudur and G. Lanckriet, "On the convergence of the concave-convex procedure," *Advances in neural information processing systems*, vol. 22, pp. 1759–1767, 2009.
[34] D. Wipf and S. Nagarajan, "A new view of automatic relevance determination," in *Adv. Neural Inf. Process. Syst.*, vol. 20, 2008.
[35] M. Tobenkin, I. Manchester, J. Wang, A. Megretski, and R. Tedrake, "Convex optimization in identification of stable non-linear state space models," in *49th IEEE Conference on Decision and Control*, pp. 7232–7237, 2010.
[36] A. Auslender and M. Teboulle, "Interior gradient and proximal methods for convex and conic optimization," *SIAM Journal on Optimization*, vol. 16, p. 697, 2006.
[37] R. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.
[38] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2008.
[39] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, no. 1-3, pp. 193–228, 1998.
[40] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial methods in convex programming*, vol. 13 of *Studies in Applied Mathematics*. Philadelphia, PA: SIAM, 1994.
[41] E. D. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Mathematical Programming*, vol. 95, no. 2, pp. 249–277, 2003.
[42] M. S. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: A Python package for convex optimization, version 1.1.5," 2012. Available at `http://abel.ee.ucla.edu/cvxopt`.
[43] M. S. Andersen, J. Dahl, Z. Liu, and L. Vandenberghe, "Interior-point methods for large-scale cone programming," in *Optimization for Machine Learning* (S. Sra, S. Nowozin, and S. J. Wright, eds.), pp. 55–83, MIT Press, 2011.
[44] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, pp. 773–795, 1995.
[45] D. Wipf and B. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3704 –3716, 2007.
[46] L. Ljung, *System Identification Toolbox for use with* MATLAB. Natick, MA: The MathWorks, Inc, 8th ed., 2012.
[47] H. Hjalmarsson and F. Gustafsson, "Composite modeling of transfer functions," *IEEE Transactions on Automatic Control*, vol. 40, no. 5, pp. 820–832, 1995.
[48] T. Chen and L. Ljung, "Implementation of algorithms for tuning parameters in regularized least squares problems in system identification," *Automatica*, vol. 49, no. 7, pp. 2213–2220, 2013.
[49] T. Chen, A. Chiuso, G. Pillonetto, and L. Ljung, "Rank-1 kernels for regularized system identification," in *IEEE 52nd Conference on Decision and Control*, pp. 5162–5167, Dec. 2013.

Tianshi Chen was born in China in November 1978. He received his M.E. degree from Harbin Institute of Technology in 2005, and Ph.D. from Chinese University of Hong Kong in December 2008. From April 2009 to March 2011, he was a postdoctoral researcher at the Automatic Control group of Linköping University, Sweden. His research interests are mainly within the areas of system identification, statistical signal processing, and nonlinear control theory and applications. He is currently an Assistant Professor at Linköping University.

Martin S. Andersen received his M.S. in Electrical Engineering from Aalborg University, Denmark (2006), and his Ph.D. in Electrical Engineering from the University of California, Los Angeles (2011). He was a postdoctoral researcher from 2011 until 2012 in the Division of Automatic Control at Linköping University, Sweden, and he is currently a postdoctoral researcher at the Technical University of Denmark in the Department of Applied Mathematics and Computer Science. His research interests include optimization, numerical methods, signal and image processing, systems and control.

Lennart Ljung received his PhD in Automatic Control from Lund Institute of Technology in 1974. Since 1976 he is Professor of the chair of Automatic Control In Linköping, Sweden. He has held visiting positions at Stanford and MIT and has written several books on System Identification and Estimation. He is an IEEE Fellow, an IFAC Fellow and an IFAC Advisor. He is as a member of the Royal Swedish Academy of Sciences (KVA), a member of the Royal Swedish Academy of Engineering Sciences (IVA), an Honorary Member of the Hungarian Academy of Engineering, an Honorary Professor of the Chinese Academy of Mathematics and Systems Science, and a Foreign Associate of the US National Academy of Engineering (NAE). He has received honorary doctorates from the Baltic State Technical University in St Petersburg, from Uppsala University, Sweden, from the Technical University of Troyes, France, from the Catholic University of Leuven, Belgium and from Helsinki University of Technology, Finland. In 2002 he received the Quazza Medal from IFAC, and in 2003 he received the Hendrik W. Bode Lecture Prize from the IEEE Control Systems Society, and he was the 2007 recipient of the IEEE Control Systems Award.

Alessandro Chiuso is Associate Professor with the Dept. of Information Engineering, University of Padova. He received the "Laurea" degree summa cum laude in Telecommunication Engineering from the University of Padova in July 1996 and the Ph.D. degree in System Engineering from the University of Bologna in 2000. He has held visiting positions with Washington University St. Louis (USA), KTH (Sweden) UCLA (USA). Dr. Chiuso serves or has served as member of several conference program committees and technical committees. He is an Associate Editor of Automatica (2008-), European Journal of Control (2011-). He was an Associate Editor of IEEE Trans. on Automatic Control (2010-2012), the IEEE Conference Editorial Board (2004-2009) and a member of the editorial board of IET Control Theory and Application (2007-2012). His research interest are mainly in Estimation, Identification Theory and Applications. Further information can be found at the personal web page `http://automatica.dei.unipd.it/people/chiuso.html`

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. In 2000 and 2002 he was visiting scholar and visiting scientist, respectively, at the Applied Physics Laboratory, University of Washington, Seattle. From 2002 to 2005 he was Research Associate at the Department of Information Engineering, University of Padova. Since 2005 he has been Assistant 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.