Multiple Kernel Based Regularized System Identification with SURE Hyper-parameter Estimator

Hong, Shiying; Mu, Biqiang; Yin, Feng; Andersen, Martin Skovgaard; Chen, Tianshi

Published in:
I F A C Workshop Series

Link to article, DOI:
10.1016/j.ifacol.2018.09.083

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Hong, S., Mu, B., Yin, F., Andersen, M. S., & Chen, T. (2018). Multiple Kernel Based Regularized System Identification with SURE Hyper-parameter Estimator. I F A C Workshop Series, 51(15), 13-18. DOI: 10.1016/j.ifacol.2018.09.083

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-15 (2018) 13–18

Multiple Kernel Based Regularized System Identification with SURE Hyper-parameter Estimator

Shiying Hong [*], **Biqiang Mu** [**], **Feng Yin** [*], **Martin S. Andersen** [***], **Tianshi Chen** [*]

[*] *School of Science and Engineering and Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, China*
[**] *Department of Electrical Engineering, Linköping University, Linköping, Sweden*
[***] *Department of Applied Mathematics and Computer Science, Technical University of Denmark, Copenhagen, Denmark*

Abstract: In this work, we study the multiple kernel based regularized system identification with the hyper-parameter estimated by using the Stein's unbiased risk estimators (SURE). To approach the problem, a QR factorization is first employed to compute SURE's objective function and its gradient in an efficient and accurate way. Then we propose an algorithm to solve the SURE problem, which contains two parts: the outer optimization part and the inner optimization part. For the outer optimization part, the coordinate descent algorithm is used and for the inner optimization part, the projection gradient algorithm is used. Finally, the efficacy of the proposed algorithm is demonstrated by numerical simulations.

Keywords: Linear system identification, regularization methods, hyper-parameter estimation, SURE, multiple kernel.

1. INTRODUCTION

Kernel-based regularization methods have been receiving increasing attention over the past few years in the system identification community; see Chen (2019), Chen and Pillonetto (2018), Mu and Chen (2018), Mu et al. (2017), Mu et al. (2018a), Chen et al. (2018) and Pillonetto et al. (2014) for a recent survey. One recent result is the so-called multiple kernel based regularization method introduced in Chen et al. (2014). It was shown there that the use of multiple kernels has a couple of advantages. For example, the multiple kernel can better model complicated systems than the single kernels, such as the stable spline (SS) kernel Pillonetto and Nicolao (2010), the diagonal correlated (DC) kernel and its special case tuned correlated (TC) kernel Chen et al. (2012). A key step of this method is to estimate the hyper-parameter of the multiple kernel by using the empirical Bayes (EB) method, see e.g., Carlin and Louis (1996). The EB method is currently the mostly widely used hyper-parameter estimation method for the regularized system identification. However, the EB method has the following limitations: first, it requires the Gaussian assumption on the measurement noise; second, it is shown to be not asymptotically optimal in the sense of mean square error (MSE), see Mu et al. (2018b) for details.

In this paper, we revisit the multiple kernel based regularization method by using instead the Stein's Unbiased Risk Estimator (SURE) to estimate the hyper-parameter, see e.g., Stein (1981). In contrast with the EB method, the SURE method does not require the Gaussian assumption and is asymptotically optimal in the sense of the mean square error Mu et al. (2018b), but the SURE method with multiple kernel is not a difference of convex programming problem, see e.g., Chen et al. (2014). Therefore, it is a critical problem how to solve the SURE with multiple kernel in an efficient way. To tackle this problem, a QR factorization is first employed to compute SURE's objective function and its gradient in an efficient and accurate way. Then we propose an algorithm to solve the SURE problem with multiple kernel, which contains two parts: the outer optimization part and the inner optimization part. For the outer optimization part, we use the coordinate descent method and for the inner optimization, we use the projection gradient method with Armijo rule. The coordinate descent method can guarantee the convergence of the outer optimization, and the Armijo rule can guarantee the convergence of the inner optimization. To check the efficacy of the proposed method, we test the first 500 test systems and data sets in the data-bank S1D1 in Chen et al. (2012). For the test system and data sets, the SURE

[★] This work was supported by the Thousand Youth Talents Plan funded by the central government of China, the general project funded by NSFC under contract No. 61773329 and 61603379, the Shenzhen research projects funded by the Shenzhen Science and Technology Innovation Council under contract No. Ji-20170189 and Ji-20160207, the President's grant under contract No. PF. 01.000249 and the Start-up grant under contract No. 2014.0003.23 funded by the Chinese University of Hong Kong, Shenzhen, as well as by a research grant for junior researchers funded by Swedish Research Council under contract No. 621-2014-5894, the National Key Basic Research Program of China (973 Program) under contract No. 2014CB845301, the President Fund of Academy of Mathematics and Systems Science, CAS under contract No. 2015-hwyxqnrc-mbq.

2405-8963 © 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.09.083

method with multiple kernel behaves similar to the EB method with multiple kernel and is more accurate in terms of average fits and more robust than the EB method and the SURE method both with the TC kernel.

2. REGULARIZED IMPULSE RESPONSE ESTIMATION

2.1 Problem Statement

First, we consider a discrete-time single-input-single-output (SISO) time-invariant linear (LTI) stable system:

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

where q is the shift operator so that $qu(t) = u(t+1)$, t is the time index, $y(t)$, $u(t)$ and $v(t)$ are the output, input and disturbance at time t, respectively. The transfer function $G_0(q)$ is defined as:

$$G_0(q) = \sum_{k=1}^{\infty} g_k^0 q^{-k} \tag{2}$$

where the coefficients g_k^0, $k = 1, \cdots, \infty$, form the impulse response of $G_0(q)$. Our goal is to find an estimator of the impulse response g_k^0, $k = 1, \cdots, \infty$ as well as possible based on data $\{u(t), y(t)\}_{t=1}^M$.

2.2 Regular Finite Impulse Response Model Estimation

Since the impulse response of a stable LTI system decays exponentially, it is reasonable enough to truncate the infinite impulse response at a sufficiently high order n leading to the finite impulse response (FIR) model:

$$G(q,\theta) = \sum_{k=1}^{n} g_k q^{-k}, \quad \theta = [g_1, g_2, \cdots, g_n]^T \tag{3}$$

Then system (1) can be rewritten as follows:

$$y(t) = \phi^T(t)\theta + v(t), t = n+1, \cdots, M \tag{4}$$

where $\phi^T(t) = [u(t-1), \cdots, u(t-n)]$ is the regressor and θ is called the impulse response vector.

It is often convenient to rewrite the linear regression model (4) in matrix form. To this goal, let:

$$Y = \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(M) \end{bmatrix}, \ \Phi = \begin{bmatrix} \phi_1^T(n+1) \\ \phi_2^T(n+2) \\ \vdots \\ \phi_M^T(M) \end{bmatrix}, \ V = \begin{bmatrix} v(n+1) \\ v(n+2) \\ \vdots \\ v(M) \end{bmatrix} \tag{5}$$

Here, we let $N = M - n$ for notational brevity. Then we can rewrite the linear regression model (4) in matrix form:

$$Y = \Phi\theta + V \tag{6}$$

where $Y \in \mathbb{R}^N$, $V \in \mathbb{R}^N$ and $\Phi \in \mathbb{R}^{N\times n}$. The regularized impulse response estimate $\hat{\theta}^R$ is the value that minimizes the regularized least squares criterion:

$$\begin{aligned} \hat{\theta}^R &= \arg\min_{\theta} \|Y - \Phi\theta\|_2^2 + \sigma^2\theta^T P(\eta)^{-1}\theta \\ &= (\Phi^T\Phi + \sigma^2 P(\eta)^{-1})^{-1}\Phi^T Y \end{aligned} \tag{7}$$

where $\sigma^2 > 0$ is a known noise variance, $\|\cdot\|_2$ is the Euclidean norm, I_N is the $N-$dimensional identity matrix, $P(\eta)^{-1}$ is the $n\times n$ regularization matrix, $P(\eta)$ is the kernel matrix, see Rasmussen and Williams (2006), $\eta \in \Gamma \subset \mathbb{R}^m$ is the parameter vector used to parameterize the kernel matrix $P(\eta)$ and called the hyper-parameter, and Γ is the set where we search for the hyper-parameter η.

2.3 Kernel Matrix

We can divide the design of kernel matrix $P(\eta)$ into two parts: parameterization of $P(\eta)$ by the hyper-parameter η, see e.g., Chen (2018), and hyper-parameter estimation for a given kernel structure. Many kernels have been introduced over the years, e.g., the stable spline (SS) kernel Pillonetto and Nicolao (2010) and the diagonal/correlated (DC) kernel and the tuned/correlated (TC) kernel Chen et al. (2012), the latter two of which are defined as follows:

$$\begin{aligned} &DC \quad P_{k,j}^{DC}(\eta) = c\lambda^{\frac{k+j}{2}}\rho^{|k-j|}, \eta = [c \quad \lambda \quad \rho] \\ &TC \quad P_{k,j}^{TC}(\eta) = c\min(\lambda^k, \lambda^j), \eta = [c \quad \lambda] \\ &\qquad c \geqslant 0, \quad 0 \leq \lambda < 1, \quad 0 \leq |\rho| \leq 1 \end{aligned} \tag{8}$$

where the subscript k, j denotes the (k,j) element of a matrix. Instead of using single kernels, it is also possible to use multiple kernels:

$$P(\eta) = \sum_{i=1}^{m} \eta_i P_i, \qquad \eta = [\eta_1, \ldots, \eta_m]^T, \quad \eta_i \geq 0 \tag{9}$$

where $\eta_i \geq 0$ and $P_i \in \mathbb{R}^{n\times n}$ is positive semi-definite.

2.4 Hyper-parameter Estimation

The value of the hyper-parameter η is in general unknown, and we need to estimate η based on the data. There are different ways to accomplish this goal. One effective way is the maximum likelihood method under the assumptions that θ is Gaussian with zero mean and covariance matrix $P(\eta)$ and V is Gaussian distributed with zero mean and covariance matrix $\sigma^2 I_N$, θ and V are independent. The method is also called the empirical Bayes method Carlin and Louis (1996), and can be described as follow:

$$\arg\max_{\eta\in\Gamma} p(Y|\eta) = \arg\max_{\eta\in\Gamma} N(0, \Phi P(\eta)\Phi^T + \sigma^2 I_N) \tag{10}$$

which is equivalent to:

$$\hat{\eta} = \arg\min_{\eta\in\Gamma} Y^T\Sigma(\eta)^{-1}Y + \log\det\Sigma(\eta) \tag{11}$$

where

$$\Sigma(\eta) = \Phi P(\eta)\Phi^T + \sigma^2 I_N \tag{12}$$

Another idea is to estimate the hyper-parameter using Stein's unbiased risk estimators (SURE), see e.g., Stein (1981), where the hyper-parameter η is estimated as follows:

$$\begin{aligned} \hat{\eta} = \arg\min_{\eta\in\Gamma} &\|Y - \Phi\hat{\theta}^{\mathrm{R}}(\eta)\|_2^2 \\ &+ 2\sigma^2 \operatorname{trace}(\Phi(\Phi^T\Phi + \sigma^2 P^{-1}(\eta))^{-1}\Phi^T) \end{aligned} \tag{13}$$

3. EFFICENT AND ACCURATE CALCULATION OF THE OBJECTIVE FUNCTION AND GRADIENT

When we develop iterative algorithms to solve the hyper-parameter estimation problem (13), we have to compute the objective function and its gradient at each iteration. As a result, the performance of our implementation relies on the efficient and accurate computation of the objective function and its gradient.

The objective function (13) can be rewritten as below:

$$\begin{aligned}&\sigma^4 Y^T(\Phi P(\eta)\Phi^T + \sigma^2 I_N)^{-2} Y \\ &+ 2\sigma^2 \operatorname{trace}((\Phi^T\Phi + \sigma^2 P^{-1}(\eta))^{-1}\Phi^T\Phi)\end{aligned} \tag{14}$$

1) **Computation Complexity** The matrix $\Phi P(\eta)\Phi^T + \sigma^2 I_N$ in (14) is of size $N \times N$, so the computation complexity of the objective function relies on the number of observation N, thus is $\mathcal{O}(N^3)$. Direct computation of the objective function in (14) is very expensive for a large N.
2) **Numerical Accuracy** The numerical accuracy is determined by the conditioning and the magnitude of the matrix $P(\eta)$ and $\Phi^T\Phi$. Both $P(\eta)$ and $\Phi^T\Phi$ can be ill-conditioned and have very large magnitude compared to the noise level $\sigma^2 I_n$, see Chen and Ljung (2013). We should find a numerically more accurate way to compute the objective function and its gradient.

In what follows, we use the same idea in Chen and Ljung (2013) to compute the objective function of (14). Note that with the Cholesky factorization (Golub and Van Loan, 2013, p. 262) of $P(\eta)$, namely, $P(\eta) = LL^T$. Let us begin with some reformulations of (14). By matrix inversion lemma,

$$\begin{aligned}&\sigma^4 Y^T(\Phi P(\eta)\Phi^T + \sigma^2 I_N)^{-2} Y \\ =&Y^T(I_N - \Phi L(\sigma^2 I_n + L^T\Phi^T\Phi L)^{-1}L^T\Phi^T)^2 Y\end{aligned} \tag{15}$$

On the other hand,

$$\begin{aligned}&2\sigma^2 \operatorname{trace}((\Phi^T\Phi + \sigma^2 P^{-1}(\eta))^{-1}\Phi^T\Phi) \\ =&2\sigma^2 n - 2\sigma^4 \operatorname{trace}((\sigma^2 I_n + L^T\Phi^T\Phi L)^{-1})\end{aligned} \tag{16}$$

As has been shown in Chen and Ljung (2013), for the EB method, the QR factorization can help to compute the objective function and its gradient in a more accurate and efficient way. Here, we take the same idea and compute the cost function (14) with QR factorization.

Firstly, we recall the definition of QR factorization (Chen and Ljung, 2013, p. 246-248). If $A = QR$ is a QR factorization of a full column rank $A \in \mathbb{R}^{m\times n}$, where $Q \in \mathbb{R}^{m\times m}$ is an orthogonal matrix and $R \in \mathbb{R}^{m\times n}$ is an upper triangular matrix. Moreover, if $Q_1 \triangleq Q(1:m, 1:n)$, $R_1 \triangleq R(1:n, 1:n)$, then $\operatorname{rank}(A) = \operatorname{rank}(Q_1)$ and $A = Q_1 R_1$ is called thin QR factorization of A. [1]

In order to guarantee the uniqueness of the thin QR factorization, we must have two assumption:

a) Without loss of generality, assume
$$\operatorname{rank}[\Phi \ Y] = n+1$$
b) Assume that all upper triangular matrices involved in the thin QR factorizations below have positive diagonal entries.

Noticing that L is positive definite, assume that we first perform the thin QR factorization of

$$\begin{bmatrix}\Phi L & Y \\ \sigma I_n & 0\end{bmatrix} = QR = Q\begin{bmatrix}R_1 & R_2 \\ 0 & r\end{bmatrix} \tag{17}$$

where Q is an $(N+n)\times(n+1)$ matrix whose columns are orthogonal unit vectors such that $Q^TQ = I_{n+1}$, and R is an $(n+1)\times(n+1)$ upper triangular matrix. Here, R is further partitioned into 2×2 blocks with R_1, R_2 and r being an $n\times n$ matrix, an $n\times 1$ vector and a scalar, respectively.

Now using $Q^TQ = I_{n+1}$, we can get that:

$$\sigma^2 I_n + L^T\Phi^T\Phi L = R_1^T R_1 \tag{18}$$

$$L^T\Phi^T Y = R_1^T R_2 \tag{19}$$

$$Y^T Y = R_2^T R_2 + r^2 \tag{20}$$

Therefore, (15) can be computed as

$$\begin{aligned}&Y^T(I_N - \Phi L(\sigma^2 I_n + L^T\Phi^T\Phi L)^{-1}L^T\Phi^T)^2 Y \\ =&r^2 - \sigma^2 R_2^T(R_1R_1^T)^{-1}R_2\end{aligned} \tag{21}$$

and (16) can be computed as

$$\begin{aligned}&2\sigma^2 n - 2\sigma^4 \operatorname{trace}((\sigma^2 I_n + L^T\Phi^T\Phi L)^{-1}) \\ =&2\sigma^2 n - 2\sigma^4 \operatorname{trace}((R_1^T R_1)^{-1})\end{aligned} \tag{22}$$

Moreover, the regularized least squares estimator for the a η can be computed according to:

$$\hat{\theta}^R = LR_1^{-1}R_2 \tag{23}$$

By defining $S \triangleq \Phi P(\eta)\Phi^T + \sigma^2 I_N$. (14) can now be rewritten as follows:

$$\sigma^4 Y^T S^{-2} Y - 2\sigma^4 \operatorname{trace}(S^{-1}) + 2\sigma^2 N \tag{24}$$

It can be shown that:

$$\frac{\partial\sigma^4 Y^T S^{-2} Y}{\partial P(\eta)} = -2\sigma^4 L^{-1}\{(R_1^T R_1)^{-1}R_1^{-1}R_2R_2^TR_1^{-T}\}L^{-1} \tag{25}$$

and

$$\begin{aligned}&-2\frac{\partial\sigma^4 \operatorname{trace}(S^{-1})}{P(\eta)} \\ &= 2\sigma^4 L^{-T}\{(R_1^T R_1)^{-1} - \sigma^2(R_1^T R_1)^{-2}\}L^{-1}.\end{aligned} \tag{26}$$

The derivative $\frac{\partial f(P(\eta)))}{\partial P(\eta)}$ can be written as:

$$\begin{aligned}\frac{\partial f(P(\eta))}{\partial P(\eta)} =&2\sigma^4 L^{-T}(R_1^T R_1)^{-1}\{I_n + \sigma^2(R_1^T R_1)^{-1} \\ &-R_1^{-1}R_2R_2^TR_1^{-T}\}L^{-1}\end{aligned} \tag{27}$$

According to the chain rule :

$$\triangledown_{\eta_i} f(\eta) = \frac{\partial f(P(\eta))}{\partial\eta_i} = \operatorname{trace}\left(\frac{\partial f(P(\eta))}{\partial P(\eta)}P_i^T\right) \tag{28}$$

However, it is obvious that the computation cost depends on the QR factorization, which depends on N. Notice that Φ and Y are fixed and only $P(\eta)$ varies when solving the SURE problem (13) with iterative algorithms. So we should make use of this observation to compute the QR factorization (17) in a more efficient way, in another word, to make the computational complexity independent of N.

Then, let us consider the thin QR factorization of

$$[\Phi \ Y] = Q_d\,[R_{d1} \ R_{d2}] \tag{29}$$

where Q_d is an $N\times(n+1)$ matrix whose columns are orthogonal unit vectors such that $Q_d^TQ_d = I_{n+1}$, R_{d1} is an $(n+1)\times n$ matrix and R_{d2} is an $(n+1)\times 1$ vector.

Now consider further the thin QR factorization of

$$\begin{bmatrix}R_{d1}L & R_{d2} \\ \sigma I_n & 0\end{bmatrix} = Q_cR_c \tag{30}$$

[1] Here, we assume $m \gg n$.

where Q_c is a $(2n+1)\times(n+1)$ matrix whose columns are orthogonal unit vectors such that $Q_c^TQ_c = I_{n+1}$, R_c is an $(n+1)\times(n+1)$ upper triangular matrix.

From (29) and (30), we have

$$\begin{bmatrix}\Phi L & Y\\ \sigma I_n & 0\end{bmatrix} = \begin{bmatrix}Q_d & 0\\ 0 & I_n\end{bmatrix} Q_c R_c \tag{31}$$

Noticing the assumptions a) and b) and that L is positive definite, it follows from (17) and (31), then

$$R = R_c, \qquad Q = \begin{bmatrix}Q_d & 0\\ 0 & I_n\end{bmatrix} Q_c \tag{32}$$

In this way, we find a more efficient way to compute the objective function and the gradient:

Algorithm 1. Objective Function and Gradient.

```
Given the QR factorization (29)
Step 1 Compute $P(\eta)$
Step 2 Compute the Cholesky factorization $L$ of $P(\eta)$
Step 3 Compute $R_{d1}L$
Step 4 Compute the QR factorization (30)
Step 5 Compute $f(\eta)$ accoding to (21) and (22)
Step 6 Compute $\frac{\partial f(P(\eta))}{\partial P(\eta)}$ according to (27)
Step 7 Compute $\bigtriangledown_{\eta_i} f(\eta)$ according to (28)
```

With the aid of thin QR factorization, the computational complexity has been reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(n^3)$ due to the inversion of smaller matrices.

4. COORDINATE DESCENT WITH PROJECTION GRADIENT METHOD

In this section, we consider SURE with multiple kernel which takes the following form:

$$\begin{aligned}\hat{\eta} &= \arg\min_{\eta\in\Gamma} f(\eta)\\ &= \arg\min_{\eta\in\Gamma} \|Y - \Phi\hat{\theta}^{\mathrm{R}}(\eta)\|_2^2\\ &\quad + 2\sigma^2\,\mathrm{trace}(\Phi(\Phi^T\Phi + \sigma^2P^{-1}(\eta))^{-1}\Phi^T)\end{aligned} \tag{33}$$

where $P(\eta)$ is the kernel matrix defined by (9) and $\Gamma = \{\eta|\eta_i \geq 0, i = 1,\cdots,m\}$.

For convenience, we divide the solution to (33) into two parts: the outer optimization part and inner optimization part. For the outer optimization, we use the coordinate descent method and for the inner optimization, we use the projection gradient method.

4.1 Outer Optimization

Coordinate Descent The coordinate descent is a non-derivative approaches for minimizing differentiable functions (Bertsekas, 1999, p.149). At each iteration, the method tries to minimize the cost along only one coordinate direction. This not only simplifies the calculation of the search direction, but often also facilitates the stepsize selection. In particular, for a given η^k, the ith coordinate of η^{k+1} is determined by:

$$\begin{aligned}\eta_i^{k+1} &= \arg\min_{x\geq0} f(\eta_1^{k+1},\cdots,\eta_{i-1}^{k+1},x,\eta_{i+1}^k,\cdots,\eta_m^k)\\ &= \arg\min_{x\geq0} g(x), \qquad i = 1,\cdots,m\end{aligned} \tag{34}$$

where η_i^k is the ith element of the hyper-parameter η at the kth iteration, and $g(x)$ is introduced for brevity.

$$\begin{aligned}g(x) &\triangleq \|Y - \Phi\hat{\theta}^{\mathrm{R}}\|_2^2 + 2\sigma^2\,\mathrm{trace}(\Phi(\Phi^T\Phi + \sigma^2P^{-1}(x))^{-1}\Phi^T)\\ P(x) &= \sum_{j=1}^{i-1}\eta_j^{k+1}P_j + xP_i + \sum_{j=i+1}^{m}\eta_j^kP_j.\end{aligned} \tag{35}$$

Since the objective function f is continuously differentiable over the set Γ, this method can converge to the stationary point of (33).

Theorem 1. Consider (33). Assume that the minimum of

$$\min_{x\geq0} f(\eta_1,\cdots,\eta_{i-1},x,\eta_{i+1},\cdots,\eta_m), \quad i = 1,\cdots,m \tag{36}$$

is uniquely attained. Let $\{\eta^k\}$ be the sequence generated by the coordinate descent method. Then every limit point of $\{\eta^k\}$ is a stationary point of (33).

Proof. Since the objective function (33) is continuously differentiable over the set Γ, then the result follows from Proposition 1.1.4 (convergence of coordinate descent) in (Bertsekas, 1999, p.151).

Stopping Criterion In our paper, the stopping criterion is defined as:

$$-\nabla f(\eta^*)^T(\eta - \eta^*) \leq 0, \quad \forall\eta \geq 0 \tag{37}$$

where $\nabla f(\eta^*)$ is the gradient of $f(\eta)$ evaluated at $\eta = \eta^*$.

Theorem 2. Consider (33). If η^* satisfies (37), then it is a stationary point of (33).

Proof. The function (37) is equivalent to:

$$((\eta^* - s\nabla f(\eta^*)) - \eta^*)^T(\eta - \eta^*) \leq 0, \forall\eta\in\Gamma, s > 0. \tag{38}$$

This holds if and only if η^* is the projection of $\eta^* - s\nabla f(\eta^*)$ on Γ accoding to Projection Theorem (Bertsekas, 1999, p19), since $\eta \geq 0$ is a nonempty, closed, and convex subset of $\mathbb{R}^n$.

Algorithm 2. Coordinate Descent Algorithm for (33).

```
Choose the starting point $\eta^0 \in \Gamma$.
FOR $k = 0, 1, 2, \ldots$ do the following steps:
    FOR $i = 1, 2, \ldots, m$, solve the problem (34), i.e.
      $\eta_i^{k+1} = \arg\min_{x\geq0} f(\eta_1^{k+1},\cdots,\eta_{i-1}^{k+1},x,\eta_{i+1}^k,\cdots,\eta_m^k)$;
    END
    IF $\eta^k$ satisfies the stopping criterion (37)
      break;
    END
END
```

4.2 Inner Optimization

Projection Gradient Method Over a Convex Set We focus on solving (34) numerically. Before we discuss the projection gradient method, we firstly recall the definition of the gradient method:

$$x^{k+1} = x^k + \alpha^kd^k, \quad k = 1, 2, \cdots, \tag{39}$$

where x^k is the parameter at kth iteration, α^k is the stepsize and d^k is the direction. If $\bigtriangledown g(x^k) \neq 0$, the direction d^k is chosen such that:

$$\bigtriangledown g(x^k)^T d^k < 0.$$

On the one hand, the stepsize α^k is chosen to be positive and such that $x^k + \alpha^k d^k \geq 0$, since $x \geq 0$. If the $\bigtriangledown g(x^k) = 0$, the method stops,i.e., $x^{k+1} = x^k$ (equivalently we choose $d^k = 0$). On the other hand, the majority of the feasible gradient methods that we will consider are also descent algorithms, that is, the step size α_k is selected so that:

$$g(x^k + \alpha^k d^k) < g(x^k), \forall k.$$

Let us recall the definition of projection operator: let z be a fixed vector in $\mathbb{R}^n$ and consider the problem of finding a vector x^* in a closed convex set Γ, which is at a minimum distance from z; that is :

$$x^* = \arg\min_{x\in\Gamma} \|z - x\|^2 = \Pi_\Gamma(z) \tag{40}$$

We call $\Pi_\Gamma(z)$ the projection of z on Γ.

For our case where $x \geq 0$ forms a convex set, we can choose a feasible direction and the stepsize such that:

$$x^{k+1} = x^k + \alpha^k(\bar{x}^k - x^k) \tag{41}$$

where

$$\bar{x}^k = \Pi_{x\geq 0}(x^k - s^k \bigtriangledown f(x^k))$$

to satisfy the requirements we discussed above. For simplicity, we choose $\alpha^k = 1, \forall k$ and we have:

$$x^{k+1} = \Pi_{x\geq 0}(x^k - s^k \bigtriangledown g(x^k)) \tag{42}$$

For our case , the projection of x^k in $x \geq 0$ is:

$$\Pi_{x\geq 0}(x^k) = \max\{0, x^k\}. \tag{43}$$

Stepsize Selection The Armijo rule is one way to choose the stepsize and can guarantee the convergence of the conditional gradient method over a convex. In particular, we choose δ, β, and $\bar{s}$, with $\bar{s} > 0$, $0 < \beta < 1$, and $0 < \delta < 1$. For each k, we set $s^k = \beta^{m_k}\bar{s}$, where m_k is the first nonnegative integer $m = 1, \cdots$ for which

$$g(x^k) - g(x(\beta^{m_k}\bar{s})) \geq \delta\beta^m\bar{s} \bigtriangledown g(x^k)^T(x^k - x(\beta^{m_k}\bar{s})) \tag{44}$$

where

$$x(\beta^{m_k}\bar{s}) \triangleq \Pi_{x\geq 0}(x^k - \beta^{m_k}\bar{s} \bigtriangledown g(x^k)) \tag{45}$$

In other words, the stepsizes $\beta^m\bar{s}$, $m = 0, 1, \cdots$ are tried successively until the above inequality is satisfied for $m = m_k$.

Algorithm 3. Projection Gradient Algorithm for (34) .

```
Choose the parameters $\beta, \delta \in (0,1)$, $\bar{s} > 0$ and starting point $x^0 \geq 0$:
FOR $k = 0, 1, 2, \ldots$ do the following steps:
    Step1. Set $m = 0$:
    Step2. Backtracking loop:
        IF (44) holds;
          THEN go to Step 3;
        ELSE
          Set $m = m + 1$ and go to the beginning of Step 2;
        END
    Step 3. Set $x^{k+1} = \Pi_\Gamma(x^k - \beta^m\bar{s} \bigtriangledown g(x^k))$;
    IF $x^{k+1}$ is stationary point;
        break;
    END
END
```

Theorem 3. Consider (34). For every $x \geq 0$ there exists $s > 0$ such that:

$$g(x) - g(x(s)) \geq \delta \bigtriangledown g(x)^T(x - x(s)) \tag{46}$$

where $x(s) \triangleq \Pi_{x\geq 0}(x - s \bigtriangledown g(x))$. Let $\{x^k\}$ be a sequence generated by the gradient projection method with the stepsize s^k chosen by the Armijo rule along the projection arc. Then every limit point of $\{x^k\}$ is stationary.

Proof. Since $g(x)$ is continuously differentiable over set $x \geq 0$, the result follows from Proposition 3.3.1 in (Bertsekas, 1999, p.283).

5. NUMERICAL SIMULATION

5.1 Description of Test Systems and Data Sets

To test Algorithm 2 and Algorithm 3, we use the data-bank S1D1 in Chen et al. (2012). S1D1 contains 2500 randomly generated 30th order discrete-time systems and associated data sets. The system is simulated with an input that is white Gaussian noise with unit variance, and the noise free output is perturbed by additive white Gaussian noise whose variance is one tenth of the variance of the noise-free output.

5.2 Simulation Setup

For this preliminary study, we only test the first 500 systems and data sets of S1D1. For each data set, we estimate an FIR model with $n = 125$ by the regularized least squares method.

For comparison, we test four methods with different kernels and different ways to estimate the hyper-parameter, which are summarized as follows:

- SURE-TC : the TC kernel (8) with the hyper-parameter estimated by the SURE method (13)
- ML-TC: the TC kernel (8) with the hyper-parameter estimated by the maximum likelihood method (11)
- ML-MK: the multiple kernel (9) with the hyper-parameter estimated by the maximum likelihood method (11)
- SURE-MK-QR: the multiple kernel (9) with the hyper-parameter estimated by the SURE method (13) of QR factorization implementation
- SURE-MK-SF: the multiple kernel (9) with the hyper-parameter estimated by the SURE method (13) of straightforward implementation

where the multiple kernel is constructed based on 22 TC kernels (8) obtained on the grid with $c = 1$ and $\lambda = 0.50 : 0.02 : 0.90, 0.95$. For ML-MK, the algorithm proposed in Chen et al. (2014) is used. For SURE-MK-QR, we use the proposed Algorithms 2 and 3, where we set $\bar{s} = 1$, $\delta = 0.01$, $\beta = 0.1$ and the initial point $\eta^0 = [0.1, 0.1, \cdots, 0.1]^T$.

Note that we estimate the noise variance σ^2 in (34) by using the sample variance of an FIR model, which is estimated with least squares method.

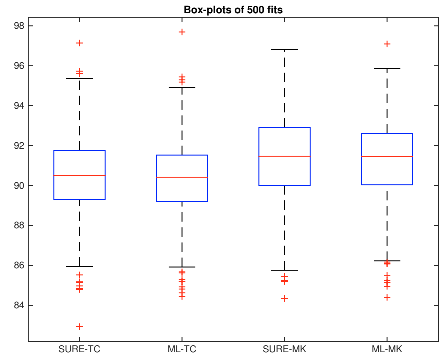

Fig. 1. Box-plots of the 500 fits.

5.3 Simulation Result

To measure the quality of regularized impulse response estimators, we define the model fit as follows:

$$fit = 100\left(1-\left[\frac{\sum_{k=1}^{125}|g_k^0-\hat{g}_k|^2}{\sum_{k=1}^{125}|g_k^0-\bar{g}^0|^2}\right]^{\frac{1}{2}}\right), \bar{g}^0 = \frac{1}{n}\sum_{k=1}^{125} g_k^0 \quad (47)$$

For each test system and associated data sets, we first calculate the regularized impulse estimators using the four methods. Then we calculate the corresponding fits (47). The average fits and times are shown in Table 1 and the distribution of the fits is shown in Figure 1.

Table 1. Average fit and average time

	ML-TC	SURE-TC	ML-MK	SURE-MK-QR	SURE-MK-SF
average fit	90.3475	90.4805	91.2777	91.3710	91.3710
average time	3.2304	3.1872	6.0372	11.3672	25.0851

5.4 Findings

For the 500 test system and data sets, SURE method with multiple kernel is more accurate in terms of average fits and more robust in contrast with the other two methods based on TC kernel. The average fit is slightly better than the multiple kernel (9) with the hyper-parameter estimated by the maximum likelihood method (11) while robustness is similar. In comparison with the straightforward implementation, the implement with QR factorization can save more than half of the computation time.

6. CONCLUSION

In this contribution, we have considered the multiple kernel based regularized system identification with the hyperparameter estimated by the Stein's unbiased risk estimator (SURE). In particular, we first perform a QR factorization on the data and then we propose a coordinate descent projection gradient algorithm, which is guaranteed to converge to a stationary point. Numerical simulation shows that the SURE with multiple kernel behaves quite well for regularized system identification. Actually, the computation time can be further shortened by using Newton's method. This will be shown in the journal version of this paper.

REFERENCES

Bertsekas, D.P. (1999). *Nonlinear programming*. Athena scientific, Belmont, Massachusetts.

Carlin, B.P. and Louis, T.A. (1996). *Bayes and Empirical Bayes methods for data analysis.* Chapman & Hall, London.

Chen, T., Andersen, M.S., Ljung, L., Chiuso, A., and Pillonetto, G. (2014). System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques. *IEEE Transactions on Automatic Control*, (11), 2933–2945.

Chen, T., Andersen, M.S., Mu, B., Yin, F., Ljung, L., and Qin, S.J. (2018). Regularized lti system identification with multiple regularization matrix. In *The 18th IFAC Symposium on System Identification (SYSID).*

Chen, T. and Ljung, L. (2013). Implementation of algorithms for tuning parameters in regularized least squares problems in system identification. *Automatica*, 49, 2213–2220.

Chen, T., Ohlsson, H., and Ljung, L. (2012). On the estimation of transfer functions, regularizations and Gaussian processes - Revisited. *Automatica*, 48, 1525–1535.

Chen, T. (2018). On kernel design for regularized LTI system identification. *Automatica*, 90, 109–122.

Chen, T. (2019). Continuous-time DC kernel — a stable generalized first-order spline kernel. *IEEE Transactions on Automatic Control.*

Chen, T. and Pillonetto, G. (2018). On the stability of reproducing kernel hilbert spaces of discrete-time impulse responses. *Automatica.*

Golub, G.H. and Van Loan, C.F. (2013). *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 4th edition.

Mu, B., Chen, T., and Ljung, L. (2018a). Asymptotic properties of generalized cross validation estimators for regularized system identification. In *The 18th IFAC Symposium on System Identification (SYSID).*

Mu, B. and Chen, T. (2018). On input design for regularized LTI system identification: Power-constrained input. *Automatica, revised in January 2018, available from http://arxiv.org/abs/1708.05539.*

Mu, B., Chen, T., and Ljung, L. (2017). On the input design for kernel-based regularized LTI system identification: Power-constrained input. *Pro. 56th IEEE Conference on Decision and Control.*

Mu, B., Chen, T., and Ljung, L. (2018b). On asymptotic properties of hyperparameter estimators for kernel-based regularization methods. *Automatica.*

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3), 657–682.

Pillonetto, G. and Nicolao, G.D. (2010). A new kernel-based approach for linear system identification. *Automatica*, 46(1), 81–93.

Rasmussen, C.E. and Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, MA.

Stein, C.M. (1981). Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 1135–1151.