An On-line Method for Estimation of Piecewise Constant Parameters in Linear Regression Models

Soheil Salehpour, Thomas Gustafsson, Andreas Johansson

E-mail: {soheil, tgu, andreas.johansson}@ltu.se, Control Engineering Group, Luleå University of Technology, SE-971 87 Luleå, Sweden.

Abstract: We present an on-line method for detecting changes and estimating parameters in AR(X) models. The method is based on the assumption of piecewise constant parameters resulting in a sparse structure of their derivative. To illustrate the algorithm and its performance, we apply it to the change in the model parameters of some ARX models. The example illustrates that the new method shows good performance for an AR(X) model with abrupt changes in the parameters.

Keywords: Parameter estimation, ARX model, LASSO, l_1 -norm, sparsity.

1. INTRODUCTION

The area of change detection is a quite active field, both in research and applications. Faults occur in almost all systems, and one aim with change detection is to locate the fault occurrence in time and raise an alarm. Another application is the estimation of perturbations (Salehpour and Johansson, 2008).

In (Gustafsson, 2000) and (Kay, 1998), surveys are given over on-line and off-line formulations of single and multiple change point estimation. In the on-line method, multiple filters are used in parallel, where each one is matched to a certain assumption on the abrupt changes. Two off-line strategies are also proposed, one is based on Markov Chain Monte Carlo techniques, and the other approach is based on a recursive local search scheme. In (Salehpour, Johansson and Gustafsson 2009) an off-line method based on MILP (Mixed Integer Linear Programming) and the sparsity of the derivative of parameters is presented which is an efficient method for fault detection and model quality estimation, but the disadvantage of this method is the computational complexity of the MILP-optimization.

An off-line LASSO (Least Absolute Shrinkage and Selection Operator) estimator is also a good choice to maximize the sparsity of $\Delta\theta(t) = \theta(t+1) - \theta(t)$, and estimate $\theta(t)$, which is used in (Salehpour and Johansson, 2008) for estimation of perturbations, in (Ozay, Sznajder, Lagoa and Camps, 2008) for set membership identification and image segmentation, and is modified to use for segmentation in (Ohlsson, Ljung and Boyd, 2010).

The goal with change point estimation, is to find a sequence $\mathbf{k}^n = [k_1, k_2, \dots, k_n]$ of time indices, where both the number n and the locations k_i are unknown, such that the signal or the model of the signal can be described as piecewise constant, where time-varying parameters are mostly constant with abrupt changes. For this purpose,

we assume that the signal can be described with a linear regression model

$$y(t) = \phi(t)^T \theta(t) + e(t) \quad (1)$$

where $\theta(t)$ is a piecewise constant vector between the time indices k^n , and $e(t)$ is some noise signal. For an ARX (n_a, n_b, n_c) model

$$\begin{aligned} \phi(t)^T &= [-y_{t-1}, \dots, -y_{t-n_a}, u_{t-n_k}, \dots, u_{t-n_k-n_b+1}] \\ \theta(t)^T &= [a_t^1, a_t^2, \dots, a_t^{n_a}, b_t^1, b_t^2, \dots, b_t^{n_b}] \\ \Theta(N) &= [\theta(1), \theta(2), \dots, \theta(N)] \end{aligned}$$

In Section 2, a LASSO estimator is described. An on-line method based on LASSO method is presented in Section 3. Simulation results are given in Section 4, followed by some concluding remarks and directions for future work in Section 5.

2. PRELIMINARIES

2.1 Estimation of Time-varying Parameters

The RLS algorithm is traditionally used as an on-line method to estimate parameters in (1), where we get $\theta(N)$

$$\hat{\theta}(N) = \underset{\theta}{\operatorname{argmin}} \sum_{t=1}^N \beta(N, t) (y(t) - \phi(t)^T \theta)^2 \quad (2)$$

where the "forgetting factor" $\beta(N, t)$ describes one of the following data windowing choices

- a1 Infinite window with $\beta(N, t) = 1$ for time-invariant signals with proper initialization.
- a2 Exponentially decaying window with $\beta(N, t) = \beta^{N-t}$, and $0 \ll \beta < 1$. RLS gives less weight to old samples and can track time-varying signals.

a3 Finite window with $\beta(N, t) = 1$ if $N - t \leq M - 1$ and $\beta(N, t) = 0$ otherwise, where the most recent M samples are used to estimate $\theta(t)$ and the rest are discarded.

A sparse matrix is defined as a matrix populated primarily with zeros. The concept of sparsity is useful in complex systems and many application areas such as network theory. Huge sparse matrices often appear in science or engineering when solving partial differential equations.

One common approach to seeking a sparse description of $\Delta\theta(t)$ is based on l_1 -norm regularization (Boyd and Vandenberghe, 2007) where most parameters are shrunk to zero. The regularized method is

$$\underset{\Theta(t)}{\text{minimize}} J(N, \Theta(t)) \quad (3)$$

where

$$J(N, \Theta(t)) = \sum_{t=1}^N \beta(N, t) (y(t) - \phi(t)^T \theta(t))^2 + \lambda_N \sum_{t=2}^N |\theta(t) - \theta(t-1)|_1$$

where λ_N is a positive parameter.

An iterative re-weighting is used in (Fazel, 2002) to get fewer parameter changes and better estimation of them. The regularization term in Eq. (3) is replaced with

$$J(N, \Theta(t)) = \sum_{t=1}^N \beta(N, t) (y(t) - \phi(t)^T \theta(t))^2 + \lambda_N \sum_{t=2}^N \omega(t) |\Delta\theta(t)|_1 \quad (4)$$

where $\omega(t) > 0$. The weights $\omega(t)$ tends to allow for successively better estimation of the nonzero coefficient locations. The algorithm is as follows:

- Set the iteration count l to zero and $w^{(l)}(t) = 1$ for $t = 2, \dots, N$.
- Solve the weighted l_1 -minimization problem, and compute $\Theta^{(l)}(t)$.
- Update the weights

$$\omega_i^{(l+1)}(t) = 1/(\epsilon + |\Delta\theta_i^{(l)}(t)|) \quad (5)$$

The largest $\Delta\theta_i(t)$ is most likely to be identified as nonzero. Once these locations are identified, their influence is down weighted in order to increase sensitivity and tending to zero for identifying the remaining small but nonzero $\Delta\theta_i(t)$.

The optimization problem with cost function (4) is solved off-line and in purpose to solve it on-line, a regularized RLS is considered

$$\hat{\theta}(N) = \underset{\theta}{\text{argmin}} J(N, \theta) \quad (6)$$

If the constant terms in $J(N, \theta)$ of (4) are neglected, the cost function can be rewritten as

$$J(N, \theta) = \frac{1}{2} \theta^T R_N \theta - \theta^T r_N + \bar{\lambda}_N \left| \theta - \hat{\theta}(N-1) \right|_1 \quad (7)$$

where

$$R_N = \sum_{t=1}^N \beta(N, t) \phi(t) \phi^T(t), \quad r_N = \sum_{t=1}^N \beta(N, t) y(t) \phi(t) \quad (8)$$

R_N and r_N can be updated recursively with different data windowing (a1-a3) as

a1: $R_N = R_{N-1} + \phi(N) \phi^T(N)$, $r_N = r_{N-1} + y(N) \phi(N)$
a2:

$$R_N = \beta R_{N-1} + \phi(N) \phi^T(N), \\ r_N = \beta r_{N-1} + y(N) \phi(N)$$

a3:

$$R_N = R_{N-1} + \phi(N) \phi^T(N) - \phi(N-M) \phi^T(N-M), \\ r_N = r_{N-1} + y(N) \phi(N) - y(N-M) \phi(N-M)$$

and $\bar{\lambda}_N(t) = \omega(t) \lambda_N$ is chosen to satisfy the oracle properties which is discussed in Section 2.2.

2.2 Adaptive LASSO and Oracle Conditions

For the asymptotic analysis of (1), two conditions are assumed in (Knight and Fu, 2000):

- 1 $e(t)$ are independent identically distributed random variables with mean 0 and variance σ^2 .
- 2 $\frac{1}{N} \phi^T \phi \rightarrow C$, where C is a positive definite matrix.

We now define the adaptive LASSO. Suppose that $\hat{\alpha}$ is a consistent estimator to α^* , and define the weight vector $\hat{\mu} = 1/|\hat{\alpha}|$. The adaptive LASSO estimates $\hat{\alpha}^{(*N)}$ is given by

$$\hat{\alpha}^{(*N)} = \underset{\alpha}{\text{argmin}} \sum_{t=1}^N \beta(N, t) (y(t) - \phi(t)^T \alpha(t))^2 + \lambda_N \sum_{t=1}^N \mu(t) |\alpha(t)|_1$$

Let $\mathcal{A} = \{j : \alpha_j^* \neq 0\}$ be a vector with length p_0 . The estimated parameters is denoted as $\hat{\alpha}_j(\delta)$, we call δ an oracle procedure (Fan and Li, 2001) if $\hat{\alpha}_j(\delta)$ has the following oracle properties:

- Identifies the right subset model, $\mathcal{A}_N^* = \{j : \hat{\alpha}_j(\delta) \neq 0\}$
- Has the optimal estimation rate and convergence in distribution ($\rightarrow_d$), that is $\sqrt{N}(\hat{\alpha}_j(\delta) - \alpha_j^*) \rightarrow_d N(0, \Sigma^*)$, where Σ^* is the covariance matrix knowing the true subset model.

It is shown in (Zou, 2006) that with a proper choice of λ_N , the adaptive LASSO enjoys the oracle properties.

Theorem (Oracle properties): Suppose that $\lambda_N/\sqrt{N} \rightarrow 0$ and $\lambda_N \rightarrow \infty$. Then the adaptive LASSO satisfies the following:

- (1) Consistency in variable selection:

$$\lim_N P(\mathcal{A}_N^* = \mathcal{A}) = 1$$

- (2) Asymptotic normality:

$$\sqrt{N}(\hat{\alpha}_{\mathcal{A}}^N - \alpha_{\mathcal{A}}^*) \rightarrow_d N(0, C_{11}^{-1})$$

and C_{11} is a $p_0 \times p_0$ submatrix of matrix

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

in the second condition of asymptotic analysis.

Let $\mathcal{B} = \{j : \Delta\theta_j^* \neq 0\}$ and $\mathcal{B}_N^* = \{j : \Delta\hat{\theta}_j(\delta) \neq 0\}$, where $\Delta\hat{\theta}_j(\delta)$ is a δ oracle procedure of $\Delta\theta_j$. Then the oracle properties can also be shown for (4).

(1) Consistency in variable selection:

$$\lim_N P(\mathcal{B}_N^* = \mathcal{B}) = 1$$

(2) Asymptotic normality:

$$\sqrt{N}(\Delta\hat{\theta}_B^N - \Delta\theta_B^*) \rightarrow_d N(0, C_{11}^{-1})$$

The proof of this is a straightforward modification of the proof in (Zou, 2006).

3. ONLINE METHOD BASED ON AN ONLINE (CYCLIC) COORDINATE DESCENT

A gradient-based minimization of (7) is impossible because the l_1 -norm is non-differentiable. A possible approach is offered by On-line coordinate descent iterative minimizers (Angelosante, Bazerque and Georgios, 2010). The algorithm is modified here to develop an online solver of (7) and compute a closed-form solution per iteration.

In cyclic coordinate descent (CCD), iterative minimization of $J(N, \theta)$ in (7) is performed with respect to one coordinate per iteration cycle. If the solution at time N and iteration i is denoted as $\theta^{(i)}(N)$, the p th variable at the i th iteration is updated as

$$\theta_p^{(i)}(N) = \underset{\theta}{\operatorname{argmin}} J(N, [\theta_1^{(i)}(N), \dots, \theta_{p-1}^{(i)}(N), \theta, \theta_{p+1}^{(i-1)}(N), \dots, \theta_{n_a+n_b}^{(i-1)}(N)]) \quad (9)$$

for $p = 1, \dots, n_a + n_b$. In every i th cycle, each coordinate p is optimized, while the pre-coordinates $(1, \dots, p-1)$ are kept fixed to their values at the i th cycle, and the post-coordinates $(p+1, \dots, n_a + n_b)$ are kept fixed to their values at $(i-1)$ th cycle. The algorithm is solvable in closed form with an effective initialization (all-zero vector), and recent comparative studies show that the complexity of the method is similar to the state-of-art-batch LASSO solvers (Wu and Lange, 2008).

An adaptive equivalent of CCD LASSO is introduced (Angelosante, Bazerque and Georgios, 2010) as online coordinate descent (OCD) algorithm to iteratively solve (9), where the iteration index (i) is replaced in OCD by the time index N , and the difficulty of OCD is to update only one variable in one direction per time.

Let $N = k(n_a + n_b) + p$ denote the time index, where $p \in \{1, \dots, n_a + n_b\}$ is the only entry of θ to be updated at time N (only θ_p is updated and $\theta_q(N) = \theta_q(N-1)$ for $q \neq p$ is kept unchanged). $k = \lfloor \frac{N}{n_a+n_b} \rfloor - 1$ is the number of cycles and how many times the p th coordinate is updated.

Let $\hat{\theta}(N-1)$ denote the solution of the OCD algorithm at time $N-1$ and $\hat{\theta}_q(N) = \hat{\theta}_q(N-1)$ for $q \neq p$, which sets all

Algorithm 1: OCD

Initialize with $\theta(0) = 0$
for $k = 0, 1, \dots$
 for $p = 1, \dots, n_a + n_b$
 1. Get data $y(N)$, and $\phi(N)$, $N = k(n_a + n_b) + p$.
 2. Compute r_N and R_N in a1-a3.
 3. Set $\theta_q(N) = \theta_q(N-1)$ for all $q \neq p$.
 4. Compute $r_{N,p}$ in (11).
 5. Update $\theta_p(N)$ as in (12).
 end for
end for

Table 1. OCD Algorithm

but the p th coordinate at time N equal to those at time $N-1$, and select the p th one by minimizing $J(N, \theta)$ as

$$\hat{\theta}_p(N) = \underset{\theta}{\operatorname{argmin}} J(N, [\hat{\theta}_1(N-1), \dots, \hat{\theta}_{p-1}(N-1), \theta, \hat{\theta}_{p+1}(N-1), \dots, \hat{\theta}_{n_a+n_b}(N-1)]) \quad (10)$$

After isolating $\theta_q(N-1)$ for $q \neq p$ in the cyclic update (10), the $J(N, \theta)$ depends on the p th coordinate, and can be rewritten as

$$\hat{\theta}_p(N) = \underset{\theta}{\operatorname{argmin}} \left[\frac{1}{2} R_N(p, p) \theta^2 - r_{N,p} \theta + \bar{\lambda}_N |\theta - x| \right] \quad (11)$$

$$r_{N,p} = r_N(p) - \sum_{q \neq p} R_N(p, q) \hat{\theta}_q(N-1)$$

where $x = \hat{\theta}_p(N-1)$. It is a scalar optimization problem, and has the closed-form solution (Friedman, Hastie, Höfling and Tibshirani, 2007)

$$\hat{\theta}_p(N) = \frac{\operatorname{sgn}(r_{N,p} - R_N(p, p)x)}{R_N(p, p)} (|r_{N,p} - R_N(p, p)x| - \bar{\lambda}_N)_+ + x \quad (12)$$

where $[\gamma]_+ := \max(\gamma, 0)$. A soft-thresholding operation sets inactive entries to the previous value, and gives a sparse solution. The OCD algorithm is shown in Table 1.

The OCD solver has low complexity but exhibits slow convergence because each variable is updated every $n_a + n_b$ observation. We implement the OCD cyclically to update all coordinates once or several times per observation. Once (10) is solved the p th coordinate will be computed by minimization in the next steps

$$\hat{\theta}_p^{(i)}(N) = \underset{\theta}{\operatorname{argmin}} J(\hat{\theta}_1^{(i)}(N), \dots, \hat{\theta}_{p-1}^{(i)}(N), \theta, \hat{\theta}_{p+1}^{(i-1)}(N), \dots, \hat{\theta}_{n_a+n_b}^{(i-1)}(N)) \quad (13)$$

$\hat{\theta}_p^{(i)}(N)$ is solved as in (12) with $x = \hat{\theta}_p^{(i-1)}(N-1)$

$$\hat{\theta}_p^{(i)}(N) = \frac{\operatorname{sgn}(\bar{r}_{N,p} - R_N(p, p)x)}{R_N(p, p)} (|\bar{r}_{N,p} - R_N(p, p)x| - \bar{\lambda}_N)_+ + x \quad (14)$$

where

$$\bar{r}_{N,p} = r_N(p) - \sum_{q < p} R_N(p, q) \hat{\theta}_q^{(i)}(N) - \sum_{q > p} R_N(p, q) \hat{\theta}_q^{(i-1)}(N) \quad (15)$$

The online cyclic coordinate descent (OCCD) is shown in Table 2.

Algorithm 2: OCCD

Initialize with $\theta(0) = 0$

for $N = 1, \dots$

1. Get data $y(N)$, and $\phi(N)$.

2. Compute $\mathbf{r}_N$ and $\mathbf{R}_N$ in $a1 - a3$.

for $l = 1, \dots$ (times to update the weights $\omega(t)$ in (5))

for $i = 1, \dots$ (times OCCD update all coordinates)

for $p = 1, \dots, n_a + n_b$

3. Compute $\bar{r}_{N,p}$ in (15).

4. Update $\theta_p(N)$ as in (14).

end for

end for

end for

end for

Table 2. OCCD Algorithm

4. SIMULATION RESULTS

In purpose to give some idea about the performance of the method, we apply it to a number of AR(X) models.

We take $\lambda_{N,max} = \sqrt{2\sigma^2 \log(n_a + n_b) \sqrt{\sum_{n=1}^N \beta^{2(N-n)}}$, for $a2$, $\lambda_{N,max} = \sqrt{2\sigma^2 \log(n_a + n_b) \sqrt{N^{0.5}}}$ for $a1$ and $a3$, $\lambda_N = 0.1\lambda_{N,max}$, and $\epsilon = 0.01$ in (5). The input is a ± 1 PRBS (Pseudo-Random Binary Sequence) signal in Examples 1 and 2.

4.1 Example 1:

The method is applied to an ARX change model with $n_a = 2$ and $n_k = n_b = 1$ and $\sigma^2 = 0.1$, where the parameters are shown in Fig. 1(a)-1(c). The input and output are depicted in Fig. 1(d)-1(e). Let $\beta = 0.9$ for $a2$, and the size of window $M = 20$ for $a3$ in the OCD and OCCD methods. We consider 5 times coordinate updating for the OCCD method. The parameter estimation for a set of data (y_t, u_t, e_t) is shown in Fig. 1(a)-1(c).

In Fig. 2, an unbiased variance of b_t^1 is also shown to compare with the RLS method for 100 data sets using Monte Carlo analysis of 10000 samples, where the a_t^1 and a_t^2 are respectively 1.5 and 0.8 and b_t^1 is changed abruptly after 5000 samples with magnitude 2. A smaller unbiased variance is obtained compared with the RLS method by using the OCCD algorithms. The RLS algorithm with window size of $M = 20$ and the OCCD algorithm are compared in Fig. 2, which shows less unbiased variance of the OCD algorithms than RLS in $a2$ with $\beta = 0.9$ and $a3$ with $M = 20$.

We also check the OCD and OCCD algorithms while b_t^1 is changed as a ramp function (Fig. 3(c)), which shows a good tracking of parameters in Fig. 3(a)-3(c), despite the parameter is not piecewise constant. Fig. 3(d)-3(e) depicts the output and the PRBS input.

4.2 Example 2:

Changing time delay: Consider the system

$$y(t) = -0.9y(t-1) + u(t-n_k) + e(t)$$

At time $t = 100$ the time delay n_k changes from 2 to 1. An ARX-model

$$y(t) = -ay(t-1) + b_1u(t-1) + b_2u(t-2) + e(t)$$

is used to estimate a , b_1 and b_2 . The OCCD method estimates the parameters with $\beta = 0.9$ for $a2$ and $M = 20$ for $a3$, which is shown in Fig. 4(a)-4(c) for a set of data (y_t, u_t, e_t) . The result shows a good estimation of b_1 and b_2 which jump with magnitude 1 at sample 50.

4.3 Example 3:

The algorithm investigates the data of a human EEG signal (Fig. 5(a)). A second order AR model is considered to model the time-varying EEG signal. An estimated and a smoothed piecewise constant parameter estimate is obtained using a bank of 8 filters from (Gustafsson, 2000). With an AR2 model, the change point 430 is computed (Fig. 5(b)). Our algorithm is implemented with $\sigma^2 = 1$ and $\beta = 0.99$. Fig. 5(b) shows that the change of EEG is detected after 20 samples and the parameter estimate converges to the estimate of the filter bank.

5. CONCLUSIONS AND FUTURE WORK

An on-line LASSO algorithm, is presented to estimate piecewise constant parameters in linear regression models. It is based on the assumption of piecewise constant parameters resulting in a sparse structure of their derivative, and a cyclic coordinate descent iterative minimization of LASSO problem. In particular, the parameters of an AR(X) model are considered. The method is tested on a linear AR(X) change model. The results shows good performance of the method.

For future research, a faster convergence of the OCCD algorithm should be pursued.

ACKNOWLEDGEMENTS

The authors wish to thank the Hjalmar Lundbohm Research Center (HLRC) funded by LKAB for financing this research.

REFERENCES

- Angelosante D., Bazerque A. and Georgios B. Online adaptive Estimation of Sparse Signals: where RLS meets the l_1 -norm. *IEEE Transactions on signal Processing*, Vol. 58, No. 7, July 2010.
- Boyd S. and Vandenberghe L. *Convex Optimization*. Cambridge University, 2007
- Fan T. and Li R. Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties *Journal of the American Statistical Association*, Vol. 96, No. 456, 2001.
- Fazel M. Matrix Rank Minimization with Applications. *PhD thesis*. Elec. Eng. Dept, Stanford University, March 2002.
- Friedman J., Hastie T., Höfling H., and Tibshirani R. Pathwise coordinate optimization *Annals of Applied Statistics*, Vol. 1, No. 2, 302-332, 2007.
- Gustafsson F. Adaptive filtering and change detection, John Wiley and Sons, Ltd, 2000.
- Kay S. M. Fundamentals of Statistical Signal Processing: Detection Theory. Prentice-Hall, 1998.
- Knight K. and Fu W. J. Asymptotics for LASSO-type estimators. *The Annals of Statistics*, Vol. 28, No. 5, 1356-1378, 2000.

Fig. 1. The ARX change model with a white Gaussian noise ($\sigma^2 = 0.1$). The true parameters (solid), OCD algorithm (dashed) with $\beta = 0.9$, OCCD algorithm with 5 times coordinates updating and $\beta = 0.9$ (dash-dotted), and OCCD algorithm with the size of windows $M = 20$ (dotted) in (a) a_t^1 , (b) a_t^2 and (c) b_t^1 (d) The output (e) Input

Fig. 2. Unbiased variance of b_t^1 , RLS (solid), RLS with windows size $M = 20$ (dash-dotted), OCCD algorithm with 5 times coordinate updating and $\beta = 0.9$ (dotted), and OCCD algorithm with the size of windows $M = 20$ (dashed)

- Ohlsson H., Ljung L. and Boyd S. Segmentation of ARX-models using sum-of-norms regularization, *Automatica*, (46), 6, 1107-1111, 2010.
- Ozay N., Sznajder M., Lagoa C. and Camps, O. A sparsification approach to set membership identification of a class of affine hybrid systems. *In Proceedings of the 47th IEEE conference on decision and control*, 123-130, Dec. 2008.
- Salehpour S., Johansson A. and Gustafsson T. Parameter estimation and change detection in linear regression models using mixed integer linear programming *Proceedings of the 15th IFAC Symposium on System Identification (SYSID)*, Saint-Malo, France, July 2009.
- Salehpour S. and Johansson A. Two Algorithms for Model Quality Estimation in State-Space Systems with Time-Varying Parameter Uncertainty, *In Proceedings of the American Control Conference*, 4809-4814, Seattle, USA, June 2008.
- Wu T. T. and Lange K. Coordinate descent algorithms for LASSO penalized regression. *Annals of Applied Statistics*, Volume 2, Number 1, 224-244, 2008.
- Zou H. The Adaptive LASSO and its Oracle Properties. *Journal of the American Statistical Association*, Vol. 101, No. 476, 2006.

Fig. 3. The ARX change model with a white Gaussian noise ($\sigma^2 = 0.1$). The true parameters (solid), OCD algorithm (dashed) with $\beta = 0.9$, OCCD algorithm with 5 times coordinate updating and $\beta = 0.9$ (dash-dotted), and OCCD algorithm with the size of windows $M = 20$ (dotted) in (a) a_t^1 , (b) a_t^2 and (c) The parameter b_t^1 as a ramp function (d) The output (e) Input

Fig. 4. The delay model with a white Gaussian noise ($\sigma^2 = 0.1$). The true parameters (solid), OCD algorithm (dashed) with $\beta = 0.9$, OCCD algorithm with 5 times coordinate updating and $\beta = 0.9$ (dash-dotted), and OCCD algorithm with the size of windows $M = 20$ (dotted) in (a) a_t^1 , (b) b_t^1 and (c) b_t^2

Fig. 5. (a) The human EEG signal (b) The estimated parameters of an AR(2) model, the estimated (dotted) and the smoothed estimation (dashed) using the bank of filters, the OCCD algorithm with 5 coordinate updates and $\beta = 0.99$ (solid)